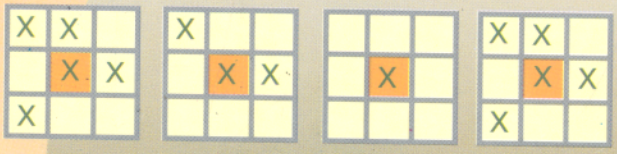
TỦ SÁCH

TIN HỌC TRONG NHÀ TRƯỜNG

BÙI VIỆT HÀ

LẬP TRÌNH PASCAL

TẬP BA



NHÀ XUẤT BẢN GIÁO DỤC

BÙI VIỆT HÀ

LẬP TRÌNH PASCAL

TẬP BA

(Tái bản lần thứ ba)

NHÀ XUẤT BẢN GIÁO DỤC

Chịu trách nhiệm xuất bản :

Chủ tịch HĐQT kiêm Tổng Giám đốc NGÔ TRẦN ÁI

Phó Tổng Giám đốc kiêm Tổng biên tập NGUYỄN QUÝ THAO

Biên tập :

NGÔ ÁNH TUYẾT

Trình bày bìa :

TẠ TRỌNG TRÍ

Chế bản :

TRẦN BÍCH VÂN

---

04 - 2006 / CXB / 194 - 1860 / GD Mã số: 8H786T6 - DAI

LỜI NHÀ XUẤT BẢN

Trong những năm gần đây, việc đào tạo về Công nghệ Thông tin (CNTT) đã trở thành nhu cầu ngày càng cấp thiết hơn đối với mọi người, từ nhân viên bán hàng, kĩ sư, bác sĩ và phóng viên, đến nhà doanh nghiệp, cán bộ quản lí... Phù hợp với xu thế phát triển của xã hội, tin học đã trở thành một môn học được yêu thích ở nhà trường. Ngay từ trên ghế nhà trường không ít học sinh đã quyết tâm học tập để trở thành chuyên gia CNTT và một số em đã đạt thành tích cao tại các kì thi học sinh giỏi trong nước và quốc tế về tin học.

Để đáp ứng nhu cầu về sách tham khảo tin học cho học sinh phổ thông và góp phần nâng cao chất lượng đào tạo tin học trong nhà trường, Nhà xuất bản Giáo dục thành lập tủ sách " Tin học trong nhà trường" và sẽ lần lượt cho ra mắt bạn đọc loạt sách được chọn lọc có hệ thống cho học sinh phổ thông.

Chúng tôi rất mong nhận được ý kiến đóng góp của các thầy cô giáo, các em học sinh và đông đảo bạn đọc gần xa để ngày càng đáp ứng tốt hơn nhu cầu của bạn đọc.

Thư từ góp ý xin gửi về NXB Giáo dục, 81 Trần Hưng Đạo, Hà Nội.

Nhà xuất bản Giáo dục

LỜI NÓI ĐẦU

Đây là quyển sách thứ ba và cuối cùng của bộ sách Lập trình Pascal dành cho học sinh phổ thông. Trong tuyển tập này, các bài tập được chia theo các chủ đề khoa học và ứng dụng khác nhau, do đó sẽ không có phần tóm tắt lý thuyết như hai tập đầu của bộ sách. Học sinh được ôn luyện theo hướng củng cố các kiến thức đã có và áp dụng vào thực tế. Hầu hết các bài tập trong sách đều thuộc phần nâng cao, chuyên sâu, đặc biệt là các bài từ 11 đến 15. Tài liệu này cũng có thể dành cho các lớp chuyên Tin hoặc dùng để luyện cho học sinh giỏi và đội tuyển Tin học. Tuy nhiên như đã nêu trong lời giới thiệu của tập đầu tiên, bộ sách Lập trình Pascal này nhằm vào đối tượng là tất cả các bạn trẻ yêu thích Tin học, yêu thích lập trình, tự mình muốn khám phá, tìm tòi những điều mới lạ của môn học đầy hấp dẫn này.

Sách bao gồm khoảng 250 bài tập, chia theo những chủ đề khác nhau. Phần cuối cùng là tuyển tập một số đề thi học sinh giỏi quốc gia và quốc tế. Do phải hạn chế dung lượng sách, chúng tôi chỉ có thể cung cấp lời giải chi tiết cho phần lớn các bài có số thứ tự lẻ.

Hi vọng rằng bộ sách sẽ có ích cho các thầy cô giáo, các em học sinh và góp phần vào việc luyện thi học sinh giỏi Tin học.

Chúng tôi rất mong nhận được những ý kiến đóng góp để bộ sách được hoàn thiện hơn trong những lần xuất bản tới.

Tác giả

PHẦN I.
TÓM TẮT LÝ THUYẾT VÀ ĐỀ BÀI TẬP

Bài 1. Thủ tục và hàm

A. Tóm tắt lý thuyết

1. Cấu trúc của một chương trình con

a) Hàm

```
Function Tên_hàm(TS1:kiểu; TS2:kiểu; ...):kiểu;
  Var    {khai báo các biến cục bộ};
  Begin
    {Các lệnh tính toán};
    Tên_hàm:= giá trị; {hàm kết thúc phải trả lại giá trị nào đó}
  End;
```

* Cách sử dụng hàm: Lời gọi hàm có dạng:

Tên_hàm(danh sách tham số thực sự)

là thành phần của các biểu thức hoặc nằm ở các vị trí được phép của các biểu thức (chẳng hạn như ở vế phải của lệnh gán).

b) Thủ tục

```
Procedure Tên_thủ_tục(TSGT1:kiểu; TSGT2:kiểu; ... ;
                      Var TSB1:kiểu; Var TSB2:kiểu; ...);
  Var    {khai báo các biến cục bộ của thủ tục};
  Begin
      {Các lệnh tính toán của chương trình con};
  End;
```

* Cách gọi thủ tục: Câu lệnh gọi thủ tục có dạng:

Tên_thủ_tục(danh sách tham số thực sự);

c) Sự khác nhau cơ bản giữa hàm và thủ tục

- Hàm cho một giá trị thông qua tên hàm và hàm có thể tham gia vào các biểu thức tính toán, thủ tục không cho một giá trị nào. Trong thân hàm bao giờ cũng có giá trị gán cho tên hàm.
- Tham số hình thức và tham số thực sự:
  - Các tham số khác nhau sau tên hàm và thủ tục trong khai báo hàm và thủ tục gọi là các *tham số hình thức* (hay còn được gọi là *đối*). Trong thủ tục các tham số hình thức có các loại: các tham số khai báo sau từ khoá Var gọi là các *tham số biến*; các tham số mà đằng trước không có từ khoá Var gọi là các *tham số giá trị*. Trong hàm chỉ có tham số giá trị.
  - *Tham số thực sự* là các tham số dùng trong lời gọi hàm hay thủ tục. Danh sách tham số thực sự trong lời gọi phải tương ứng với danh sách tham số hình thức trong khai báo chương trình con, kể cả về kiểu.
- Trong thủ tục, các tham số giá trị thường là các biến chứa dữ liệu đưa vào thủ tục, các tham số biến là các biến chứa kết quả tính toán của thủ tục khi ra khỏi thủ tục, ta có thể dùng chúng để tính toán tiếp.

2. Biến toàn cục, biến cục bộ

Biến toàn cục là biến được khai báo ở đầu chương trình chính, tồn tại trong suốt thời gian chạy của chương trình. Ta có thể sử dụng và làm thay đổi giá trị của biến toàn cục nhờ các câu lệnh trong chương trình chính cũng như trong các chương trình con.

Biến cục bộ là biến được khai báo ở đầu một chương trình con. Chúng được cấp phát bộ nhớ khi chương trình con được gọi tới và được giải phóng khi ra khỏi chương trình con. Biến cục bộ có hiệu lực trong chương trình con và tất cả các chương trình con khác nằm trong chương trình con này.

Nếu tên biến cục bộ của một chương trình con trùng với tên biến toàn cục thì máy cũng không nhầm lẫn, nó sẽ dùng hai ô nhớ khác nhau để lưu trữ hai biến, khi ra khỏi chương trình con thì biến cục bộ bị xoá.

3. Cách truyền dữ liệu

Khi gặp một lời gọi tới một chương trình con, máy sẽ thực hiện cac công việc sau:

- Cấp phát bộ nhớ cho các đối, các biến cục bộ.

- Truyền giá trị của các tham số thực sự cho các tham số giá trị tương ứng, truyền địa chỉ của các tham số thực sự ứng với tham số biến cho các tham số biến của thủ tục.
- Thực hiện các lệnh trong chương trình con, trong khi thực hiện chương trình con, các biến cục bộ và tham số giá trị có thể bị biến đổi nhưng không ảnh hưởng tới các biến bên ngoài. Nhưng mọi thay đổi của tham số biến trong chương trình con sẽ kéo theo sự thay đổi của các tham số thực sự tương ứng (vì ở đây là truyền theo địa chỉ). Do đó, khi ra khỏi chương trình con, các tham số thực sự ứng với các tham số biến vẫn giữ được giá trị mới nhất do chương trình con tạo ra.
- Thực hiện xong các lệnh của chương trình con, máy sẽ giải phóng các đối và các biến cục bộ, trở về nơi gọi chương trình con.

Lấy kết quả thực hiện chương trình con: nếu là hàm thì lấy giá trị kết quả ở tên hàm, nếu là thủ tục thì lấy kết quả ở các tham số thực sự ứng với các tham số biến. Khi cần lấy duy nhất một giá trị từ chương trình con thì ta dùng hàm, còn nếu không cần lấy giá trị nào hoặc cần từ hai giá trị trở nên thì ta dùng thủ tục.

B. Câu hỏi và bài tập

1. Tìm lỗi trong các hàm sau:

a.
```
Function f(a: 'a'..'z'): integer;
Begin
        f:=ord(a)-ord('p');
        if f<0 then f:=-f;
End;
```

b.
```
Function g(k: integer): 0..maxint;
Var i,s: 0..maxint;
Begin
        s:=0;
        for i:=1 to k do s:=s+sqrt(i);
End;
```

c.
```
Function h(x: integer): integer;
Begin
        h(x):=(sqr(x)+x)/2;
End;
```

2. Cho hàm số:

```
Function fib(n: integer): integer;
```

```
Begin
        if n<3 then fib:=1
        else fib:=fib(n-1)+fib(n-2);
End;
```

Hãy tính nhẩm fib(2), fib(3) và fib(5).

3. Trong 4 hàm số sau đây có 1 hàm đúng dùng để tính n!. Bạn hãy tìm hàm số đó:

- ```
  Function fac(n: integer): integer;
  Begin
          fac:=n*fac(n-1);
  End;
  ```

- ```
  Function fac(n: integer): integer;
  Begin
          if n=0 then fac:=1
           else fac:=fac(n+1)/(n+1);
  End;
  ```

- ```
  Function fac(n: integer): integer;
  Begin
          if n=0 then fac:=1
           else fac:=n*(n-1)*fac(n-2);
  End;
  ```

- ```
  Function fac(n: integer): integer;
  Begin
          if n=0 then fac:=1
           else fac:=n*fac(n-1);
  End;
  ```

4. Cho hàm số:

```
Function f(n: integer): integer;
Begin
        if n>100 then f:=n-10
         else f:=f(f(n+11));
End;
```

Hãy tính f(106), f(99) và f(85). Tìm công thức tổng quát cho f(n).

5. Bạn hãy viết hàm số tính C(m, n) theo công thức sau:

$C(0, k) = C(k, k) = 1$ với mọi $0 \le k \le n$,

$C(m, n) = C(m, n-1) + C(m-1, n-1)$ khi $0 < m < n$.

6. Bạn hãy kiểm tra xem các chương trình sau đây sẽ in ra kết quả gì:

- ```
  Program Print;
  Var x,y: char;
  Procedure P(x: integer);
  Const y=true;
  Begin
          writeln(x,' ',y);
  End;
  Procedure Q;
  Var x: char;
  Begin
          x:=succ(y); y:='*';
          writeln(x,' ',y);
  End;
  BEGIN
          x:='a'; y:='5';
          P(8);
          writeln(x,' ',y);
          Q;
          writeln(x,' ',y);
  END.
  ```
- ```
  Program Print;
  Type Str=String[5];
  Var i: integer; t: Str;
  Procedure P(var s: Str);
  Begin
          i:=1;
          while s[i]<'9' do
            begin
                s[i]:=succ(s[i]);
                i:=i+1;
            end;
  End;
  BEGIN
  ```

```
        i:=1; t:='12945';
        P(t);
        writeln(t[i]);
END.
```

7. Cho hàm F(n) với n tự nhiên xác định như sau:

$$F(0) = 0, F(1) = 1, F(2n) = F(n), F(2n+1) = F(n) + F(n+1).$$

Với N cho trước, lập chương trình không dùng mảng độ dài N, tính F(N).

8. Bạn hãy dẫn ra một vài ví dụ trong Pascal, khi mà các biểu thức sau không thoả mãn:

$$y = y; \quad y + y = 2*y.$$

9. Một file dữ liệu chứa các phân số, mỗi phân số gồm tử số và mẫu số là 2 số nguyên chứa trong 2 trường tương ứng của một bản ghi. Bạn hãy viết một chương trình tìm phân số tối giản của các phân số đó, kết quả ghi ra một file khác có cấu trúc giống file dữ liệu vào. Yêu cầu trong chương trình phải dùng một hàm tính USCLN của 2 số và một thủ tục giản ước một phân số.

10. Mỗi phân số gồm tử số và mẫu số là 2 số nguyên được chứa trong 2 trường tương ứng của một record. Bạn hãy viết một chương trình nhập 2 phân số và 1 dấu phép tính, sau đó tính kết quả. Yêu cầu trong chương trình phải dùng một hàm tính USCLN của 2 số và các thủ tục tính tổng, hiệu, tích, thương của 2 phân số.

11. Cho trước toạ độ các đỉnh của một đa giác. Bạn hãy tính chu vi và đường kính của nó. (Đường kính ở đây hiểu là khoảng cách lớn nhất giữa hai đỉnh của đa giác). Trong chương trình sử dụng hàm tính khoảng cách giữa 2 đỉnh. Dữ liệu được nhập từ tệp POLYGON.DAT có dạng: dòng đầu tiên ghi số n, n dòng tiếp theo ghi tọa độ phẳng của các đỉnh đa giác. Kết quả đưa ra màn hình.

12. Cho trước đa giác lồi $A_1A_2A_3...A_n$ (tức là cho toạ độ các đỉnh tương ứng của nó). Bạn hãy lập chương trình tính diện tích của nó. Trong chương trình sử dụng các hàm tính khoảng cách giữa 2 đỉnh và diện tích tam giác. Dữ liệu nhập vào giống bài trên.

13. Mỗi ma trận vuông cấp n các số thực $A = (a_{ij})$ có tương ứng một số detA gọi là định thức của nó cho bởi công thức truy hồi sau:

$$\det A = \begin{cases} a_{11} & khi\ n=1 \\ (-1)^2 a_{11} \det A_{11} + (-1)^3 a_{12} \det A_{12} + ... + (-1)^{n+1} a_{1n} \det A_{1n} & khi\ n>1 \end{cases}$$

trong đó A_{1i} là ma trận nhận được từ A bằng cách bỏ đi hàng thứ i, cột thứ 1. Bạn hãy lập chương trình tính định thức của một ma trận cho trước. Dữ liệu ma trận cho trong tệp IN.DAT.

14. Cho trước đa thức hệ số nguyên $f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$. Các nghiệm nguyên của đa thức (nếu có) phải là các ước số của a_0. Bạn hãy lập chương trình tìm nghiệm nguyên của f(x). Trong chương trình sử dụng hàm tính giá trị đa thức theo sơ đồ Hoorner. Số n và dãy số a_1, a_2, ..., a_n được nhập từ bàn phím.

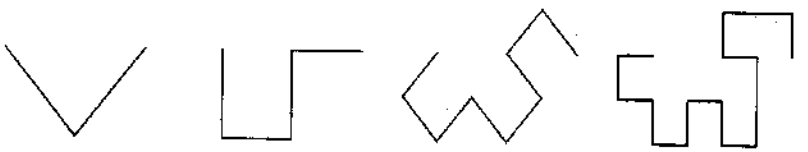
15. Mỗi ma trận vuông cấp n các số thực $A = (a_{ij})$ với $\det A \neq 0$ có tương ứng một ma trận $B = (b_{ij})$ cùng cấp gọi là ma trận nghịch đảo của nó cho bởi công thức:

$$b_{ij} = \frac{(-1)^{i+j} \det A_{ij}}{\det A},$$

trong đó A_{ij} nhận được từ A bằng cách bỏ đi hàng thứ i, cột thứ j.
Lập chương trình tính ma trận nghịch đảo của ma trận A. Input và Output của bài toán ghi trong các tệp có tên INPUT.DAT và OUTPUT.DAT.

16. Hãy lập chương trình để xếp 12 con mã lên bàn cờ tướng sao cho mỗi ô trên bàn cờ bị đúng một con mã kiểm soát.

17. Đường gấp khúc "con rồng" là đường có dạng như sau:



Hình trên vẽ 4 con rồng cấp 1, 2, 3 và 4 tương ứng. Hãy sử dụng kĩ thuật đệ qui vẽ con rồng cấp n (n nhập từ bàn phím).

Bài 2. Làm việc với file

A. Tóm tắt lý thuyết

1. Các thao tác cơ bản với tệp

- Khai báo kiểu tệp:

biến_tệp = **File Of** Kiểu_phần_tử;

Ví dụ

```
Type    FileByte = File Of Byte;

Var f1 : FileByte;
    f2 : File Of Char;
```

- Mở tệp mới để lưu dữ liệu:

  Assign(biến_tệp, tên_tệp);
  ReWrite(biến_tệp);

Ghi dữ liệu vào tệp thông qua thủ tục Write:

  Write(biến_tệp, biến1, biến2, ..., biếnN);

Trong đó: biến1, biến2, ..., biếnN chứa các giá trị cần ghi vào tệp, có thể là các hằng, biến, biểu thức.

- Mở tệp để đọc dữ liệu

  Assign(biến_tệp, tên_tệp);
  Reset(biến_tệp);

Đọc dữ liệu từ một tệp thông qua thủ tục Read:

  Read(biến_tệp, biến1, biến2, ..., biếnN);

Trong đó: biến1, biến2, ..., biếnN sẽ chứa các giá trị cần đọc từ tệp, có thể là các hằng, biến, biểu thức.

- Đóng tệp: Sau khi đọc dữ liệu từ một tệp hay cất dữ liệu mới vào một tệp nên dùng thủ tục đóng tệp để đảm bảo thông tin trên tệp là đầy đủ và tin cậy, thủ tục đóng tệp có cú pháp:

  Close(biến_tệp);

- Có thể truy nhập trực tiếp vào một phần tử của tệp thông qua thủ tục SEEK, có cú pháp như sau:

  SEEK(biến_tệp, Stt);

Trong đó: Stt là số thứ tự của phần tử trong tệp (phần tử đầu tiên của tệp đánh số là 0). Gặp lệnh này máy sẽ đặt con trỏ tệp vào phần tử thứ Stt để ta thực hiện các thao tác cần thiết ở vị trí đó.

2. Các hàm và thủ tục xử lí tệp

+ **Erase**(biến_tệp); xoá tệp trên đĩa có tên ấn định với biến_tệp. Chú ý là không được xoá tệp đang mở.
+ **Rename**(biến_tệp, S); dùng để thay đổi tên tệp bằng tên mới kiểu String chứa trong xâu S. Tên mới không được trùng với các tên tệp nào đó đã có sẵn trên đĩa đang làm việc. Không được đổi tên tệp đang mở.
+ **FileSize**(biến_tệp); cho số phần tử của tệp biến_tệp. Hàm nhận giá trị bằng 0 khi tệp rỗng.
+ **FilePos**(biến_tệp); cho biết vị trí hiện tại của con trỏ tệp (phần tử đầu tiên của tệp là phần tử số 0). Muốn ghi thêm một phần tử vào cuối tệp trước hết ta phải đưa con trỏ tệp về vị trí cuối cùng của tệp qua lệnh sau:

SEEK(biến_tệp, FileSize(biến_tệp));

3. Tệp văn bản (Text Files)

- Khai báo tệp văn bản: biến tệp f có kiểu TEXT được khai báo như sau:

Var f : TEXT;

Thành phần cơ sở của tệp kiểu TEXT là kí tự. Tệp văn bản được cấu trúc theo dòng, mỗi dòng được kết thúc bởi dấu EOLn (End Of Line), với Turbo Pascal đó là các cặp kí tự điều khiển CR (Carriage Return: nhảy về đầu dòng, mã ASCII = 13) và LF (Line Feed: nhảy xuống dòng tiếp theo, mã ASCII = 10). Tệp này được kết thúc bởi dấu EOF (End Of File), với Turbo Pascal đó là Ctrl-Z (^Z), mã ASCII = 26.

Các thủ tục đã trình bày đối với tệp cấu trúc cũng sử dụng được đối với tệp văn bản, tuy nhiên tệp văn bản cũng có những đặc điểm khác sau:

▪ Các hàm chuẩn với tệp văn bản:

EOF(var f:Text):Boolean;

Hàm này trả về giá trị True khi duyệt đến cuối tệp, ngược lại trả lại giá trị False. Ta thường dùng hàm này để kiểm tra xem đã đọc hết tệp văn bản chưa.

EOLN(var f:Text):Boolean;

Hàm này trả về giá trị False nếu duyệt chưa đến cuối dòng, ngược lại cho giá trị True. Hàm này thường dùng để kiểm tra xem đã đọc hết dòng chưa.

▪ Các thủ tục với tệp văn bản:

+ Đọc dữ liệu từ tệp văn bản đã có trên đĩa:

Read(biến_tệp,biến 1,biến 2, ..., biến N);

Readln(biến_tệp, biến 1, biến 2, ..., biến N);

Readln(biến_tệp);

Trong đó: biến 1, biến 2, ..., biến N có thể là các biến thuộc kiểu kí tự, nguyên, thực, logic, chuỗi. Lệnh Read đọc một hay nhiều phần tử mà không chuyển con trỏ tệp xuống dòng, Readln cũng đọc như Read nhưng lại chuyển con trỏ tệp xuống đầu dòng tiếp theo sau khi đã lần lượt đọc các biến tương ứng. Lệnh Readln(biến_tệp) chỉ đưa con trỏ tệp xuống đầu dòng tiếp theo mà không đọc gì cả.

+ Ghi vào một tệp văn bản:

Ghi dữ liệu vào tệp thông qua các lệnh Write, Writeln cụ thể như sau:

(1) **Write**(biến_tệp, biến 1, biến 2, ..., biến n);

(2) **Writeln**(biến_tệp, biến 1, biến 2, ..., biến n);

(3) **Writeln**(biến_tệp);

Lệnh (1) có tác dụng viết các giá trị của các biến 1, biến 2, ..., biến n (các giá trị này là các hằng hay biểu thức có kiểu đơn giản như: nguyên, thực, chuỗi, logic, kí tự) vào biến tệp.

Lệnh (2) tương tự như lệnh (1) nhưng đưa dấu hiệu hết dòng vào tệp sau khi đã ghi hết các giá trị đó.

Lệnh (3): chỉ thực hiện việc đưa thêm dấu hiệu hết dòng vào tệp.

+ Thủ tục thêm dòng vào tệp:

APPEND(var f:Text);

Lệnh Append mở tệp văn bản ra để bổ sung thêm các dòng, định vị con trỏ tệp vào cuối tệp.

B. Câu hỏi và bài tập

1. Hãy kiểm tra xem các khẳng định sau là đúng hay sai:

a. Các phần tử của file phải cùng một kiểu.

b. Kiểu file và kiểu mảng chỉ khác nhau ở chỗ: kiểu file có độ dài tuỳ ý, còn kiểu mảng có độ dài hạn định.

c. Tại một thời điểm không thể vừa đọc, vừa ghi vào cùng một file.

d. Không thể dùng phép so sánh hai file.

e. Không thể dùng phép gán cho biến kiểu file.

2. Trả lời các câu hỏi sau:

a. Có thể thêm phần tử mới vào một file được hay không? Thêm vào chỗ nào: vào đầu, vào giữa hay vào cuối file?

b. Có thể bớt một phần tử khỏi một file được hay không? Bớt được từ chỗ nào: từ đầu, từ giữa hay từ cuối file?

3. Cho một text file chứa ma trận kề của một đồ thị với qui cách như sau:

- Dòng đầu tiên chứa một số tự nhiên n là số đỉnh của đồ thị.
- n dòng tiếp theo là n dòng tương ứng của ma trận kề: mỗi dòng gồm n số nhị phân.

Bạn hãy lập chương trình kiểm tra xem đó là đồ thị có hướng hay vô hướng và lập danh sách kề cho đồ thị đó. Danh sách kề được ghi vào một text file theo qui cách:

- Dòng đầu tiên (thứ 0) chứa số tự nhiên n là số đỉnh của đồ thị.
- n dòng tiếp theo là n dòng tương ứng của danh sách kề: dòng thứ i gồm các số là các đỉnh kề với đỉnh thứ i. Các số phân cách nhau bởi dấu cách.

4. Cho danh sách kề của một đồ thị. Bạn hãy lập chương trình tạo ma trận kề cho đồ thị đó. Danh sách kề và ma trận kề được ghi trong các text file theo qui cách như bài 3.

5. Cho một text file f. Bạn hãy lập chương trình kiểm tra xem đó có phải là một văn bản đối xứng hay không.

6. Hãy khai báo kiểu của biến x sao cho biểu thức sau là đúng: x^[5].y+[5]

7. Cho một văn bản chứa trong một text file f. Trong văn bản,tính từ trái sang phải, từ trên xuống dưới, kí tự # là dấu hiệu xoá đi một kí tự đứng ngay trước nó nếu có. Ví dụ: '#Ta#ôi đi ngủ#h###học' có nghĩa là 'Tôi đi học'. Bạn hãy viết chương trình sửa lại file f theo qui ước trên.

8. Cho 2 file f1 và f2 cùng kiểu (nhưng không rõ kiểu nào). Bạn hãy lập thủ tục gán nội dung của file f2 cho file f1.

9. Cho hai file số nguyên f và g đã sắp xếp theo thứ tự tăng dần. Bạn hãy lập chương trình nhập hai file đó thành một file h cũng đã được sắp xếp. Yêu cầu chỉ đọc qua mỗi file một lần và không dùng mảng trung gian.

10. Cho hai file cùng kiểu f, g. Chỉ dùng một kiểu file h làm trung gian, hãy ghi nội dung của file f sang g và ngược lại, nội dung của file g sang file f.

11. Cho một File Text chứa danh sách học sinh lớp học, mỗi học sinh trên một dòng. Ví dụ một file như vậy:

Nguyễn Văn Bình
Trần Quang Vinh
Lê Thị Gái

Hãy sửa lại nội dung file text đó, thêm vào số thứ tự đứng trước tên mỗi học sinh. Ví dụ ta có file đã được sửa lại:

1. Nguyễn Văn Bình
2. Trần Quang Vinh
3. Lê Thị Gái

12. Cho trước 4 file cùng kiểu F1, F2, F3, F4. Chỉ dùng một biến file làm trung gian, hãy ghi lại nội dung các file trên theo sơ đồ sau:

F1 ghi sang F2
F2 ghi sang F3
F3 ghi sang F4
F4 ghi sang F1

13. Cho file F kiểu bản ghi, mỗi bản ghi là một số tự nhiên. Hãy sao chép nội dung của F sang file G nhưng với thứ tự ngược lại của các số đã ghi trong F.

14. Cho file F kiểu bản ghi, mỗi bản ghi là một số tự nhiên. Hãy sao chép nội dung của file F sang file G nhưng với điều kiện G chỉ được giữ lại những số tự nhiên khác nhau từng đôi một của F.

15. Cho một file text. Hãy viết chương trình đếm xem file text trên chứa bao nhiêu từ.
(*Chú ý*: theo qui định, các từ cách nhau bởi một hay nhiều dấu cách).

16. Cho một file text ghi lại một chương trình Pascal của một bạn học sinh. Hãy viết chương trình kiểm tra lỗi của chương trình Pascal trên theo các cách sau đây:

Cách 1: Kiểm tra xem số lượng các dấu '(' và dấu ')' có bằng nhau không?
Cách 2: Kiểm tra xem số lượng các từ Begin và End có bằng nhau không?

17. Cho một file text. Viết chương trình loại bỏ các khoảng trống thừa bên trong file text này.

Bài 3. Số nguyên và hệ đếm

Bài tập

Trong các bài tập từ 1 đến 11, dữ liệu được nhập vào từ bàn phím. Kết quả thể hiện trên màn hình.

1. Cho hai số tự nhiên m, n với $n \neq 0$. Hỏi có bao nhiêu chữ số khác nhau trong biểu diễn thập phân của phân số $\frac{m}{n}$?

2. Cho một số thập phân. Hãy đổi số đó sang dạng (xâu) nhị phân.

3. Cho một số nhị phân. Hãy đổi số đó sang dạng số thập phân.

4. Cho một số nhị phân. Hãy đổi số đó sang hệ mười sáu.

5. Cho một số nhị phân. Hãy đổi số đó sang hệ bát phân.

6. Kim giờ tạo với nửa đường thẳng qua tâm đồng hồ và số 12 một góc φ

a. Hãy tìm vị trí của kim phút.

b. Hỏi đồng hồ đang chỉ mấy giờ (chính xác đến giây)?

7. Cho biết thời gian hiện tại là h: m: s - h giờ, m phút, s giây ($0 \leq h < 12$, $0 \leq m, s < 60$). Hỏi tới khi nào thì kim giờ và kim phút gặp nhau?

8. *Liên phân số* là biểu thức dạng:

$$\cfrac{1}{b_1 + \cfrac{1}{b_2 + \cfrac{1}{\cdots\cdots \; \cfrac{1}{b_k}}}}$$

trong đó $b_1, b_2, \ldots, b_k$ là các số tự nhiên. Liên phân số trên thường được kí hiệu bằng $[b_1, b_2, \ldots, b_k]$. Mỗi phân số hữu tỉ $\frac{s}{t}$ ($0 < s < t$, s và t là các số tự nhiên) đều có thể đưa về dạng liên phân số bằng thuật toán sau:

- Chia t cho s ta được a dư r: $t = as + r$, suy ra:

$$\frac{s}{t} = \frac{1}{t/s} = \frac{1}{a + r/s}$$

- Đặt b_1 = a rồi tiếp tục biến đổi r/s...

Quá trình biến đổi dừng lại khi số dư r bằng 0.
Cho s và t. Hãy lập chương trình tìm $[b_1, b_2, ..., b_k]$.

9. Cho một số Ả rập. Hãy lập chương trình viết số đó bằng chữ số La mã.

10. Cho một số La mã. Hãy lập chương trình viết số đó bằng chữ số Ả rập.

11. Thuật toán Euclide được sử dụng để tìm ước số chung lớn nhất của hai số nguyên dương a và b, kí hiệu UCLN (a, b). Giả sử UCLN (a, b) = d thì phương trình ax+by = d có nghiệm p, q nguyên. Dùng chính dãy biến đổi Euclide để tính d, ta có thể tính được p, q.
Cho trước a, b. Hãy lập chương trình tính p và q.

12. Các số nguyên lớn (hệ thập phân) được mô tả bởi mảng số sau:

Array [0..MaxInt] of 0..9;

Hãy viết chương trình thực hiện các chức năng sau:

- Nhập số nguyên lớn từ bàn phím.
- Thực hiện các phép cộng, trừ, nhân, chia các số nguyên lớn.

13. Hãy sử dụng kết quả của bài trên để tính 2^{200}. Kết quả thể hiện dưới dạng một số nguyên lớn.

14. Hãy sử dụng kết quả của bài trên để tính 100!. Kết quả thể hiện dưới dạng một số nguyên lớn.

15. Hãy đổi một số nguyên lớn sang dạng nhị phân. Kết quả thể hiện dưới dạng một số nhị phân lớn, tức mảng kí tự : Array [0..MaxBin] of 0..1;

16. Hãy đổi một số nhị phân lớn sang dạng thập phân. Kết quả thể hiện dưới dạng một số nguyên lớn dạng thập phân.

17. Hãy viết chương trình thực hiện các chức năng sau:

- Nhập số nguyên lớn dạng nhị phân từ bàn phím.
- Thực hiện các phép cộng, trừ, nhân, chia các số nguyên lớn dạng nhị phân.

Bài 4. Đồ hoạ

A. Tóm tắt lý thuyết

1. Khởi tạo đồ hoạ

+ **InitGraph**(Gd,Gm : integer; Path : string); thủ tục chuyển màn hình từ chế độ văn bản sang chế độ đồ hoạ

Trong đó: Gd - là biến chỉ loại card màn hình.

Gm - chỉ mode màn hình.

Path - là đường dẫn đến thư mục chứa các tệp điều khiển màn hình. Nếu các tệp này thuộc thư mục hiện hành thì có thể cho Path = ''.

**** Các hàm và thủ tục liên quan đến việc khởi tạo đồ hoạ***

+ **GraphResult** : Integer; - cho biết kết quả của việc khởi tạo đồ hoạ. Nếu thành công cho giá trị 0 (hay GrOK).

+ **GraphErrorMsg**(ErrorCode : Integer): String; trả về chuỗi kí tự thông báo lỗi được xác định trong biến ErrorCode là kết quả của hàm GraphResult.

+ Thủ tục **CloseGraph** cho phép chấm dứt chế độ đồ hoạ để trở về chế độ văn bản.

2. Các thủ tục vẽ hình đơn giản

a) Vẽ điểm

+ **PutPixel**(x,y : Integer ; Clr : Word); vẽ một điểm ảnh có chỉ số màu Clr tại toạ độ (x,y) trên màn hình.

+ **GetPixel**(x, y:Integer); trả về chỉ số màu của điểm ảnh có toạ độ (x,y).

b) Vẽ đường thẳng

+ **MoveTo**(x,y); đưa con trỏ từ vị trí hiện hành tới điểm (x,y).

+ **MoveRel**(dx,dy); đưa con trỏ từ vị trí hiện hành tới điểm có tọa độ cách nó một khoảng xác định bởi dx,dy.

+ **Line**(x1,y1,x2,y2); vẽ đường thẳng từ điểm (x1,y1) đến (x2,y2).

+ **LineTo**(x,y); nối đoạn thẳng từ vị trí con trỏ hiện hành đến điểm (x,y).

+ **LinRel**(dx,dy); nối đoạn thẳng từ vị trí con trỏ hiện hành đến điểm có toạ độ cách nó một khoảng xác định bởi dx,dy; dx dương khi vẽ bên phải màn hình, dy dương khi vẽ xuống dưới màn hình.

c) Vẽ cung tròn, đường tròn, elip

+ **Arc**(x,y : Integer; SAngle, EAngle, R : Word); vẽ một cung tròn có các thành phần sau:

x, y: toạ độ tâm.

SAngle: góc bắt đầu cung (so với phương ngang).

EAngle: góc kết thúc cung.

R: bán kính của cung.

+ **Circle**(x,y : Integer ; R : Word); vẽ vòng tròn có tâm (x,y) bán kính R.

+ **Ellipse**(x,y : Integer ; SAngle,EAngle,Rx,Ry : Word); vẽ elip có các thành phần sau:

x, y: tâm của Elip.

SAngle: góc bắt đầu.

EAngle: góc kết thúc (so với phương ngang).

Rx, Ry: lần lượt là bán kính trục ngang và trục đứng của Elip.

d) Vẽ hình chữ nhật, hộp chữ nhật

+ **Rectangle**(x1,y1,x2,y2 : Integer); vẽ hình chữ nhật có toạ độ góc trên bên trái là (x1,y1) và toạ độ góc dưới phải là (x2,y2).

+ **Bar**(x1,y1,x2,y2 : Integer); vẽ hộp chữ nhật đặc.

+ **Bar3d**(x1,y1,x2,y2 : Integer; Depth : Word; Top : Boolean); vẽ khối hộp ba chiều.

Trong đó: x1,y1 : là toạ độ đỉnh trái trên của hộp;

x2,y2 : là toạ độ đỉnh phải dưới của hộp;

Depth : xác định chiều sâu của hộp;

Top : quy định hộp có nắp ở trên (True = TopOn) hay không có nắp (False = TopOff).

e) Vẽ đa giác

+ **DrawPoly**(NumPoints : Word; Var Polygon : PolyPoint);

Kiểu PolyPoint dùng để xác định toạ độ các đỉnh có cấu trúc như sau:

```
Type
        PointType = Record
                x, y : Integer;
            End;
PolyPoint = Array [1..NumPoints] of PointType;
```

3. Các thủ tục xác định màu nền, màu vẽ và tô màu hình vẽ

+ **SetPalette**(ColorNum : Word; Color : ShortInt); thủ tục định nghĩa lại màu Color cho màu thứ ColorNum trong bảng màu.

+ **SetColor**(Clr:Integer); định nghĩa màu vẽ, trong đó Clr là chỉ số màu.

+ **SetBkClolor**(Bk :Integer); thủ tục xác định màu nền.

+ **SetLineStyle**(Lt,Pt,Tk : Word); thiết lập kiểu vẽ đường thẳng.
Trong đó:

Lt: thiết lập kiểu vẽ đường, được xác định bởi các hằng đã định nghĩa như sau:

SolidLn = 0 : đường nét liền
DottedLn = 1 : đường nét đứt
CenterLn = 2 : đường xuyên tâm
DashedLn = 3 : đường nét sọc
UserBitLn = 4 : đường do người sử dụng định nghĩa.

Pt: xác định màu vẽ, tuỳ thuộc vào biến Lt.

Tk : xác định độ dày của đường, có thể lấy một trong các giá trị đã định nghĩa như sau:

NormAWidth = 1 : độ dày bình thường.
ThickWidth = 3 : nét đậm.

+ **Bar**(x1, y1, x2, y2 : Integer); vẽ hộp chữ nhật đặc có tọa độ góc trên bên trái là (x1, y1) và tọa độ góc dưới bên phải là (x2, y2). Màu và kiểu mẫu tô bên trong được thiết lập trong thủ tục SetFillStyle.

+ **SetFillStyle**(Pattern : Word ; Color : Word); thủ tục này có tác dụng cho tất cả các lệnh tô màu bên trong vật thể.

+ **GetFillPattern**(Var FP : FillPatternType); trả về định nghĩa của mẫu tô hiện hành.

+ **GetFillSettings**(Var FS : FillSettingsType); thủ tục để biết màu và mẫu tô hiện hành. Trong đó kiểu FillSettingsType được định nghĩa như sau:

```
Type
        FillSettingsType = Record
                    Pattern : Word;
                    Color : Word;
                End;
```

+ **SetFillPattern**(Pattern: FillPatternType; Color : Word); thủ tục tự thiết kế mẫu tô.

Trong đó: kiểu FillPatternType được định nghĩa như sau:

```
        Type   FillPatternType = Array[1..8] of Byte;
```

Mỗi phần tử của mảng sẽ là một số nhị phân 8 bit quy định pixel sáng - tối trên màn hình.

Biến Pattern (mẫu tô) có thể nhận giá trị từ 0 đến 12, ví dụ:

EmptyFill = 0 : Màu nền
SolidFill = 1 : Màu đặc

LineFill = 2 : Đường liền nét
LtSlashFill = 3 : Gạch chéo thưa
SlashFill = 4 : Gạch chéo dày
BkSlashFill = 5 : Gạch chéo ngược
LtBkSlashFill = 6 : Gạch chéo ngược thưa.
WideDoFill = 10 : chấm thưa
CloseDoFill = 11 : chấm dày
UserFill = 12 : Sử dụng mẫu tô màu trước đó.

+ **FillEllipse**(x,y : Integer ; Rx,Ry : Word); tô màu toàn bộ phần bên trong của elip.

+ **Sector**(x,y : Integer ; SAngle,EAngle,Rx,Ry : Word); tô màu một phần elip xác định bởi các tham số đã cho.

+ **FloodFill**(x,y :Integer ; Border : Word); tô màu một miền bất kì.

Trong đó: (x, y) là toạ độ một điểm bất kì thuộc vùng cần tô, màu tô được xác định bởi biến Border.

4. Các hàm và thủ tục xử lí đồ hoạ

+ **ImageSize**(x1,y1,x2,y2 : Integer); trả về số byte cần thiết để lưu trữ màn hình đồ hoạ giới hạn bởi hình chữ nhật (x1,y1,x2,y2).

+ **GetMem**(Var P : Point; i : integer); thủ tục tạo vùng nhớ (i bytes) quản lí bởi biến con trỏ.

+ **GetImage**(x1,y1,x2,y2 : Integer; Var Bitmap); chép các điểm nằm trong vùng màn hình giới hạn bởi (x1,y1,x2,y2) vào vùng nhớ có địa chỉ lưu trong biến bitmap (thường dùng biến động và được xác định kích thước bởi hàm ImageSize).

+ **PutImage**(x,y : Integer; Var Bitmap; CopyMode : word); chép các ảnh lưu trữ trong biến bitmap ra màn hình tại toạ độ (x,y), tham số CopyMode quy định cách thức chép ảnh: kết hợp ảnh đã có và ảnh được chép theo các phép toán logic (and, or, not, xor). Các hằng được định nghĩa trong Unit Graph:

CopyPut = 0	AndPut = 3
XorPut = 1	NotPut = 4
OrPut = 2	

5. Văn bản trên đồ họa

a) Thủ tục chọn kiểu chữ

+ **SetTextStyle**(Ft, Dr, Sz : Word); thiết lập các thông số về kiểu chữ.

Trong đó:

Ft (Font) : tên font chữ. Có các hằng đã qui định:

DefautFont = 0;
TriplexFont = 1;
SmallFont = 2;
SansSerifFont = 3;
GothicFont = 4;

Dr (Direction): hướng thể hiện chữ:

HorizDir = 0 - vẽ từ trái qua phải;
VertDir = 1 - vẽ theo hướng thẳng đứng;

Sz (Size) : kích thước của chữ mang các giá trị 1, 2, ..., trong đó đặc biệt giá trị 0 (= UserCharSize) cho phép xác định độ rộng và cao theo ý muốn của người dùng bằng cách sử dụng thủ tục:
SetUserCharSize(MulX, MulY, DivX, DivY : Word);
Trong đó: MulX, MulY : chiều rộng và chiều cao của chữ cần chọn.
DivX, DivY : chiều rộng và chiều cao của chữ thường.

b) Phân bố chữ trên màn hình

Thủ tục **SetTextJustify**(Hz,Vt : Word); cho phép phân bố chữ tương đối theo chiều ngang (Hz) và chiều dọc (Vt) của màn hình:
Hz có thể lấy các giá trị đã định nghĩa như sau:
LeftText = 0 - chữ viết nằm bên phải trục thẳng đứng đi qua con trỏ hiện hành của màn hình đồ hoạ.
CenterText = 1 - chữ viết giữa trục thẳng đứng đi qua con trỏ hiện hành.
RightText = 2 - chữ viết nằm bên trái trục thẳng đứng đi qua con trỏ hiện hành.
Vt có thể nhận các giá trị sau:
BottomText = 0 - chữ viết ở bên trên trục nằm ngang đi qua vị trí của con trỏ hiện hành.
CenterText = 1 - chữ viết ở giữa trục nằm ngang đi qua con trỏ hiện hành.
TopText = 2 - chữ viết ở dưới trục nằm ngang đi qua con trỏ hiện hành.

c) Các thủ tục xuất chuỗi ra màn hình

+ **OutText**(S : String) - xuất chuỗi S ra màn hình tại vị trí con trỏ hiện hành.

+ **OutTextXY**(x, y : Word; S : String) - xuất chuỗi S ra màn hình tại vị trí có toạ độ (x,y).

B. Bài tập

1. Vẽ một hình tam giác đều đặc quay quanh tâm của mình với vận tốc đều.

2. Vẽ một hình tam giác thường đặc quay quanh trọng tâm của mình với vận tốc đều.

3. Hãy biểu thị một dao động điều hoà của một điểm trên đường thẳng nằm ngang (công thức độ dời $x = c(1+\cos wt)/2$).

4. Biểu thị chuyển động quay đều của một hình tròn phẳng quanh một đường kính thẳng đứng của nó. Tại thời điểm t, hình tròn có dạng một elip có trục lớn R không đổi, còn trục bé $r = R.|\cos wt|$.

5. Hãy hiển thị một số hình tròn chuyển động xiên trên màn hình với hướng và vận tốc khác nhau, khi gặp cạnh màn hình sẽ phản xạ lại.

6. Chương trình sau hiển thị con trỏ chuột trên màn hình. Bạn hãy gõ và chạy thử chương trình:

```
Program Show_Text_Mouse;
Uses Dos;
Var reg: registers;
Procedure InitMouse;
Begin
  reg.ax:=$0000;
  intr($33,reg);
End;
Procedure ShowMouse;
Begin
  reg.ax:=$0001;
  intr($33,reg);
End;
BEGIN
  InitMouse;
  ShowMouse;
  Readln;
END.
```

7. Chương trình sau cũng hiển thị con trỏ chuột trên màn hình, nhưng ở chế độ đồ hoạ. Bạn hãy gõ và chạy thử chương trình:

```
Program Show_Mouse;
```

```
Uses Dos,Graph;
Var reg: registers;
    Gd,Gm: Integer;
Procedure InitMouse;
Begin
 reg.ax:=$0000;
 intr($33,reg);
End;
Procedure ShowMouse;
Begin
 reg.ax:=$0001;
 intr($33,reg);
End;
BEGIN
 Gd:=Detect;
 InitGraph(Gd,Gm,' ');
 InitMouse;
 ShowMouse;
 Readln;
 CloseGraph;
END.
```

8. Để lấy được toạ độ con trỏ chuột vào các biến XMouse, YMouse ta dùng thủ tục sau:

```
Procedure GetMousePosition(Var XMouse,YMouse: integer);
Begin
  reg.ax:=$0003;
  intr($33,reg);
  XMouse:=reg.cx;
  YMouse:=reg.dx;
End;
```

a. Sử dụng các thủ tục đã cho ở trên để lập chương trình in ra toạ độ của con trỏ chuột tại vị trí hiện thời của nó theo mẫu sau:

```
...
Repeat
 GetMousePosition(XMouse,YMouse);
 Str(XMouse,S1); Str(YMouse,S2);
 Outtextxy(XMouse,YMouse,S1+':'+S2);
```

```
Until KeyPressed;
...
```

b. Nếu in ra toạ độ con trỏ chuột theo mẫu a. thì số liệu in ra sẽ bị nhoè. Hãy cải tiến đoạn chương trình trên bằng cách xoá đi số liệu cũ trước khi in ra số liệu mới. Có thể làm điều đó bằng cách in đè lên số liệu cũ bằng chính dòng số liệu đó nhưng cùng màu với màu nền.

c. Nếu ta không di chuột mà vẫn thực hiện việc "xoá đi, ghi lại" như ở b. thì số liệu in ra trên màn hình sẽ bị nháy. Hãy khắc phục nhược điểm đó bằng cách kiểm tra xem con chuột có bị di hay không (toạ độ của nó có bị thay đổi hay không) trước khi in.

9. Trong bài trên, "bóng" của con trỏ chuột vẫn có thể làm cho số liệu in ra sẽ bị nhoè khi ta di chuột nhanh. Để khắc phục hiện tượng đó có thể "che" con trỏ chuột trước khi in bằng thủ tục sau:

```
Procedure HideMouse;
Begin
  reg.ax:=$0002;
  intr($33,reg);
End;
```

Bạn hãy sử dụng các thủ tục đã cho để lập chương trình in ra toạ độ của con trỏ chuột tại vị trí hiện thời của nó.

10. Để kiểm tra việc kích chuột, ta có thể dùng hàm MouseClick sau:

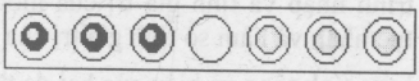
```
Function MouseClick: Char;
Var n: char;
Begin
  reg.ax:=$0003;
  intr($33,reg);
  If (reg.bx and 1)=1 then n:='L';
  If (reg.bx and 2)=2 then n:='R';
  MouseClick:=n;
End;
```

Hãy lập chương trình kiểm tra việc kích chuột:

- Nếu kích trái thì in ra toạ độ hiện thời của con trỏ chuột.
- Nếu kích phải thì dừng chương trình.

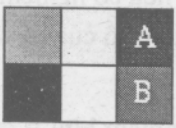
11. Hiển thị một hình vuông 10×10 ô vuông màu xanh. Khi kích chuột vào một ô vuông màu xanh thì nó sẽ đổi sang màu đỏ, khi kích chuột vào một ô vuông màu đỏ thì nó sẽ đổi sang màu xanh.

12. Hiển thị trên màn hình một bàn cờ 7 ô, trên đó có 6 quân cờ: 3 đen, 3 trắng như hình sau:



Mỗi quân cờ có thể đi vào ô trống bên cạnh hoặc vào ô trống nằm ngay sau quân bên cạnh nó. Đi một quân bằng cách kích chuột vào quân đó.

13. Hiển thị một bảng hình chữ nhật có kích thước 2×3 = 6 ô vuông, trong đó xếp 5 hình vuông màu sắc khác nhau, chỉ có 1 ô trống. Mỗi hình vuông đứng cạnh ô trống có thể trượt sang ô trống bằng cách kích chuột vào nó.



14. Hiển thị một bàn cờ tướng, trên đó xếp một con hậu. Con hậu có thể di chuyển từ ô nọ sang ô kia bằng cách kích chuột vào ô mà nó sẽ đến.

15. Hiển thị một bàn cờ tướng, trên đó xếp một con hậu và một con xe. Mỗi con cờ có thể di chuyển từ ô nọ sang ô kia bằng cách kích chuột vào nó để chọn, sau đó kích chuột vào ô mà nó sẽ đến.

16. Bạn hãy lập chương trình kiểm tra việc kích đúp chuột:

- Nếu kích đúp (trái) thì in ra toạ độ hiện thời của con trỏ chuột.
- Nếu kích phải thì dừng chương trình.

17. Bạn hãy lập chương trình kiểm tra việc rê chuột:

- Nếu rê chuột (trái) thì in ra toạ độ con trỏ chuột tại thời điểm bắt đầu và kết thúc động tác kéo rê.
- Nếu kích phải thì dừng chương trình.

18. Mô tả mô hình và chuyển động của Rubic.

Bài 5. Đa thức

Bài tập

1. Một đa thức có thể được cho dưới dạng một bản ghi gồm 2 trường: Deg chứa bậc của đa thức và Koef là mảng chứa các hệ số của đa thức theo thứ tự từ thấp đến cao.
Viết chương trình nhập và tính giá trị của một đa thức. Chương trình phải gồm thủ tục nhập và hàm số tính giá trị của một đa thức.

2. Viết các thủ tục tính tổng và hiệu của hai đa thức cho trước.

3. Viết thủ tục tính tích của hai đa thức cho trước.

4. Viết thủ tục tính luỹ thừa bậc n của một đa thức cho trước.

5. Viết thủ tục thực hiện phép chia hai đa thức cho trước (tính thương và số dư).

6. Một đa thức hệ số nguyên nếu có nghiệm hữu tỉ thì nghiệm đó phải có dạng $x = \pm \frac{k}{l}$, trong đó k là ước số của hệ số tự do, l là ước số của hệ số cao nhất.
Viết chương trình tìm các nghiệm hữu tỉ của một đa thức cho trước có hệ số nguyên.

7. Một đa thức bậc 3 hệ số nguyên nếu có 1 nghiệm x_0 là hữu tỉ thì có thể phân tích ra 2 thừa số: một thừa số bậc nhất là $(x-x_0)$ và một thừa số bậc 2 là $g(x)$.
Viết chương trình tìm 1 nghiệm hữu tỉ (nếu có) của một đa thức bậc 3 hệ số nguyên và phân tích đa thức đó thành 2 thừa số.

8. Viết chương trình tính đạo hàm của một đa thức cho trước.

9. Viết chương trình tìm nguyên hàm của một đa thức cho trước.

10. Viết chương trình khảo sát và vẽ đồ thị của một hàm đa thức bậc 3 cho trước.

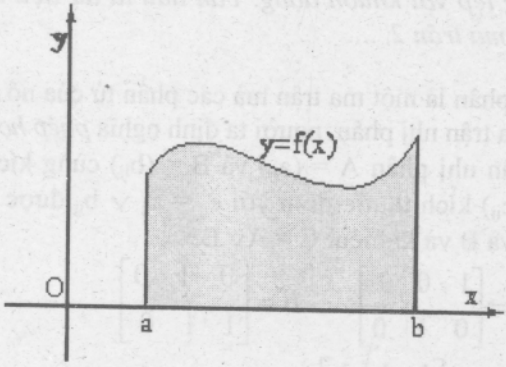
11. Bạn hãy viết chương trình khảo sát và vẽ đồ thị của một hàm đa thức bậc 4 là hàm trùng phương cho trước.

12. Bạn hãy viết chương trình tìm một đa thức có bậc cao nhất là USC của 2 đa thức cho trước (theo thuật toán Euclide).

13. Bạn hãy viết chương trình tìm một đa thức có bậc thấp nhất là BSC của 2 đa thức cho trước.

14. Cho trước đa thức P(x). Bạn hãy tìm P(x+1) - P(x).

15. Bạn hãy viết chương trình tính diện tích hình sau:



trong đó a, b là 2 số còn f(x) là một đa thức cho trước.

16. Cho trước hàm đa thức y = P(x) biểu diễn một đường cong trong hệ toạ độ Oxy. Bạn hãy tìm đa thức Y = Q(X) biểu diễn đường cong ấy trong hệ toạ độ OXY, biết x = X+a, y = Y+b.

17. Cho trước $x_0, x_1, ..., x_n$ là các nghiệm của một đa thức bậc n với hệ số cao nhất là 1. Tính các hệ số của đa thức đó.

18. Cho trước các đa thức P(x) và Q(x). Bạn hãy lập chương trình tính P(Q(x)), Q(P(x)).

19. Cho trước số $n \geq 2$. Bạn hãy viết chương trình tính các hệ số của đa thức Chebursev $T_n(x)$, biết $T_0(x) = 1$, $T_1(x) = x$, $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ với mọi $k \geq 2$.

Bài 6. Biến đổi ma trận

Bài tập

Trong tất cả các bài tập của phần này, dữ liệu ma trận đều được lưu trong các tệp văn bản với khuôn dạng sau: dòng đầu tiên ghi các số là bậc của ma trận, các dòng tiếp theo ghi dữ liệu là các hàng của ma trận. Trong trường hợp có nhiều ma trận cần nhập, các dữ liệu này được lưu trong tệp với khuôn dạng: ban đầu là dữ liệu ma trận 1, tiếp theo là dữ liệu ma trận 2, ...

1. Ma trận nhị phân là một ma trận mà các phần tử của nó hoặc là 0, hoặc là 1. Đối với ma trận nhị phân, người ta định nghĩa *phép hợp* như sau:
Cho hai ma trận nhị phân $A = (a_{ij})$ và $B = (b_{ij})$ cùng kích thước $m\times n$. Ma trận $C = (c_{ij})$ kích thước $m\times n$ với $c_{ij} = a_{ij} \vee b_{ij}$ được gọi là *hợp* của hai ma trận A và B và kí hiệu: $C = A\vee B$.

Ví dụ: Cho $A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$; $B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

thì $A\vee B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Bạn hãy lập chương trình tính hợp của hai ma trận nhị phân cho trước.

2. *Phép giao* của hai ma trận nhị phân được định nghĩa như sau:
Cho hai ma trận nhị phân $A = (a_{ij})$ và $B = (b_{ij})$ cùng kích thước $m\times n$. Ma trận $C = (c_{ij})$ kích thước $m\times n$ với $c_{ij} = a_{ij} \wedge b_{ij}$ được gọi là *giao* của hai ma trận A và B và kí hiệu $C = A \wedge B$.

Ví dụ: Cho $A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$; $B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

thì $A \wedge B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

Bạn hãy lập chương trình tính giao của hai ma trận nhị phân cho trước.

3. *Phép nhân logic* hai ma trận nhị phân được định nghĩa như sau:
Cho ma trận nhị phân $A = (a_{ij})$ kích thước $m\times k$ và $B = (b_{ij})$ kích thước $k\times n$. Ma trận $C = (c_{ij})$ kích thước $m\times n$ với $c_{ij} = (a_{i1} \wedge b_{1j})\vee (a_{i2} \wedge b_{2j})) \vee$

...$\vee$ ($a_{ik} \wedge b_{kj}$) được gọi là *tích logic* của ma trận A với ma trận B và kí hiệu C = AΘ B.

Ví dụ: Cho $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$; $B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$

thì $A\Theta B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Hãy lập chương trình tính tích logic của hai ma trận nhị phân cho trước.

4. *Luỹ thừa logic bậc r* của ma trận nhị phân vuông A, kí hiệu $A^{[r]}$, được định nghĩa như sau:

$A^{[r]} = A\Theta\, A\Theta\, A\Theta \ldots \Theta\, A$ (r thừa số A).

Ví dụ: Cho $A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

thì $A^{[0]} = E$, $A^{[1]} = A$,

$A^{[2]} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ $A^{[3]} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

$A^{[4]} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ $A^{[5]} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ ...

Hãy lập chương trình tính luỹ thừa logic bậc r của một ma trận nhị phân cho trước.

5. Ta nhắc lại một số khái niệm và định nghĩa:

- Cho ma trận A = (a_{ij}) kích thước m×n. Ma trận B kích thước n×m với $b_{ji} = a_{ij}$ được gọi là *ma trận chuyển vị* của ma trận A và kí hiệu B = A^T (hoặc A*).

- Cho hai ma trận $A = (a_{ij})$ và $B = (b_{ij})$ cùng kích thước m×n. Ma trận $C = (c_{ij})$ cùng kích thước m×n với $c_{ij} = a_{ij} + b_{ij}$ được gọi là *tổng* của hai ma trận A và B và kí hiệu C = A+B.
- Cho ma trận $A = (a_{ij})$ kích thước m×k và ma trận $B = (b_{ij})$ kích thước k×n. Ma trận $C=(c_{ij})$ kích thước m×n với $c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + a_{i3} * b_{3j} + ... + a_{ik} * b_{kj}$ được gọi là *tích* của ma trận A với ma trận B và kí hiệu C = AB.

Bạn hãy tạo các thủ tục thực hiện các công việc sau:

- Tìm ma trận chuyển vị A^T của ma trận A cho trước.
- Tính tổng C = A+B của hai ma trận A và B cho trước.
- Tính tích C = AB của ma trận A với ma trận B cho trước.
- Tìm luỹ thừa bậc n của ma trận A cho trước, n cũng cho trước.

6. *Phép nhân Tenxơ* hai ma trận số thực được định nghĩa như sau:

Cho ma trận $A = (a_{ij})$ kích thước m×n và ma trận $B = (b_{ij})$ kích thước p×q.

Ma trận $C = (c_{ij})$ kích thước mp× nq với $c_{(i-1)\,p+r\,(j-1)\,q+s} = a_{ij}b_{rs}$ được gọi là *tích Tenxơ* của ma trận A với ma trận B và kí hiệu: C = A⊗B.

Ví dụ: Cho $A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$; $B=\begin{bmatrix}0 & 1 & 5\\ -1 & 3 & 7\end{bmatrix}$

thì $A\otimes B=\begin{bmatrix}0 & 1 & 5 & 0 & 2 & 10\\ -1 & 3 & 7 & -2 & 6 & 14\\ 0 & 3 & 15 & 0 & 4 & 20\\ -3 & 9 & 21 & -4 & 12 & 28\end{bmatrix}$

Bạn hãy lập chương trình tính tích Tenxơ của hai ma trận cho trước.

7. Cho một ma trận vuông $A = (a_{ij})$ cấp n. Chọn ra n phần tử của A, trong đó không có phần tử nào cùng hàng hay cùng cột (có n! tổ hợp như vậy). Với mỗi một tổ hợp n phần tử của A chọn như trên, ta sắp xếp lại các phần tử theo thứ tự tăng dần của chỉ số thứ nhất (i). Khi đó các chỉ số thứ hai (j) lập nên một hoán vị của n số 123...n.

Ta có thể kiểm tra tính chẵn lẻ của một hoán vị theo cách sau: Hoán vị được gọi là *chẵn* nếu số các nghịch thế của nó là một số chẵn. (Trong một hoán vị của n số 123...n, *một nghịch thế* là một cặp 2 số mà số lớn đứng trước số bé. Ví dụ trong hoán vị 14325 có 3 nghịch thế: 43, 42 và

32; trong hoán vị 54321 có 10 nghịch thế: 54, 53, 52, 51, 43, 42, 41, 32, 31 và 21).

Định nghĩa dấu của tổ hợp là -1 nếu hoán vị lẻ, là +1 nếu hoán vị chẵn. Định thức của A là tổng tất cả (n!) các số, mỗi số là tích của n phần tử trong một tổ hợp chọn theo cách ở trên cùng với dấu của nó. Bạn hãy lập chương trình tính định thức của một ma trận cho trước.

8. Định thức của ma trận A có thể tính theo công thức truy hồi sau:

$$\det A = \begin{cases} a_{11} & khi\ n = 1 \\ (-1)^2 a_{11} \det A_{11} + (-1)^3 a_{12} \det A_{12} + \ldots + (-1)^{n+1} a_{1n} \det A_{1n} & khi\ n > 1 \end{cases}$$

trong đó A_{1i} là ma trận nhận được từ A bằng cách bỏ đi hàng thứ i, cột thứ 1.

Ví dụ: Với n = 2 thì $\det A = a_{11}\det A_{11} - a_{12}\det A_{12} = a_{11}a_{22} + a_{12}a_{21}$.
Với n = 3 thì $\det A = a_{11}\det A_{11} - a_{12}\det A_{12} + a_{13}\det A_{13} = \ldots$
Bạn hãy lập chương trình tính định thức của một ma trận cho trước. So sánh kết quả với bài trên.

9. Mỗi ma trận vuông cấp n các số thực $A = (a_{ij})$ với $\det A \neq 0$ có tương ứng một ma trận có cùng kích thước với A, kí hiệu A^{-1} và gọi là ma trận *nghịch đảo* của ma trận A với tính chất $A^{-1}A = AA^{-1} = E$.
Ma trận đơn vị E là ma trận (e_{ij}) được tính như sau:

$e_{ij} = 1$ nếu $i = j$ và $e_{ij} = 0$ nếu $i <> j$.

Ma trận nghịch đảo $A^{-1} = (b_{ij})$ của A có thể tính bởi công thức:

$$b_{ij} = \frac{(-1)^{j+i} \det A_{ji}}{\det A}$$

Trong đó A_{ji} là ma trận nhận được từ A bằng cách bỏ đi hàng thứ j, cột thứ i.
Bạn hãy lập chương trình tính ma trận nghịch đảo của ma trận A và kiểm tra tính chất của nó.

10. Cho một hệ phương trình bậc nhất n phương trình n ẩn:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

Lập các ma trận hệ số A, B và ma trận ẩn X:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots\dots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Khi đó hệ phương trình có thể viết dưới dạng (phép nhân) ma trận là AX = B. Nếu detA $\neq 0$ thì hệ có nghiệm duy nhất và có thể tính được theo công thức $X = A^{-1}B$. Bạn hãy lập chương trình giải hệ phương trình theo cách trên.

Chú ý: Nếu detA = 0 thì hệ sẽ vô nghiệm hoặc vô số nghiệm.

11. Cho trước ma trận A. Bạn hãy tạo các thủ tục thực hiện các công việc sau:

a. Đổi chỗ 2 hàng của ma trận đó.
b. Chia một hàng của ma trận đó cho một số khác 0.
c. Thay một hàng của ma trận đó bằng tổng của chính hàng đó với tích của một hàng khác nhân với một số.

12. Để giải hệ phương trình bậc nhất n phương trình n ẩn, ta có thể khử lần lượt các biến bằng phương pháp cộng đại số theo cách sau:

- <u>Bước 1</u>. Khử biến:
  Xét lần lượt i từ 1 (dòng thứ nhất) đến n (dòng cuối cùng), với mỗi i:
  a. Nếu $a_{ii} = 0$ thì đổi chỗ 2 phương trình thứ i và thứ j, trong đó $j > i$ và $a_{ji} \neq 0$.
  b.Chia cả hai vế cho a_{ii}.
  c. Lấy phương trình thứ j trừ đi $a_{ji} \times$ phương trình thứ i, với j chạy từ i+1 đến n.
- <u>Bước 2</u>. Tính ẩn: Xét lần lượt i từ n (dòng cuối cùng) đến 1 (dòng thứ nhất), với mỗi i tính được

$$x_i = b_i - (a_{i\,i+1}\,x_{i+1} + a_{i\,i+2}\,x_{i+2} + \dots + a_{i\,n}\,x_n).$$

Chú ý: Nếu không tồn tại j ở bước 1.a. thì hệ sẽ vô nghiệm hoặc vô số nghiệm.

Ví dụ: Giải hệ:

$$\begin{cases} 3x+6y-9z=-9 & (1) \\ x+2y+2z=2 & (2) \\ -x+4y+6z=0 & (3) \end{cases}$$

- Bước 1:

i =1: x + 2y - 3z = -3 (chia (1) cho 3)
0x - 0y + 5z = 5 (lấy (2) trừ đi (1))
0x + 6y + 3z = -3 (lấy (3) trừ đi (-1)*(1))

i =2: x + 2y - 3z = -3
0x + y + z/2 = 1/2 (đổi chỗ (2) ↔ (3), chia (2) cho 6)
0x - 0y + 5z = 5 (lấy (3) trừ đi 0*(2): không đổi)

i =3: x + 2y - 3z = -3.
0x + 0y + z/2 = 1/2
0x - 0y + z = 1 (chia (3) cho 5)

- Bước 2:

i = 3: z = 1
i = 2: y = -1/2 - z/2 = -1
i = 3: x = -3 - (2y - 3z) = 2

Vậy hệ có nghiệm x =2, y = -1, z =1.

Dùng các thủ tục đã viết ở bài 10, bạn hãy lập chương trình giải hệ phương trình theo cách trên.

Đầu vào của chương trình là ma trận:

$$C = (A,B) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots\dots & & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{bmatrix}$$

gọi là ma trận mở rộng của hệ, đầu ra là ma trận X.

13. Cho trước ma trận vuông A. Hãy tính ma trận A^2.

14. Cho trước ma trận vuông A. Hãy tính ma trận A^5.

15. Cho trước ma trận vuông A và số tự nhiên n. Hãy tính ma trận A^n một cách nhanh nhất.

16. Trên các ma trận số, người ta định nghĩa các phép Biến Đổi Sơ Cấp (BĐSC) sau đây:

BĐSC1: Hoán vị hai hàng hoặc cột
BĐSC2: Nhân một hàng hoặc một cột với một số khác không
BĐSC3: Thay đổi giá trị của một hàng hoặc một cột theo biến đổi sau đây: $C_i = C_i - \lambda C_j$
hoặc

$$H_i = H_i - \lambda H_j$$

(C_i và H_i chỉ ra các vectơ cột và hàng của ma trận)
Cho trước ma trận vuông A. Hãy sử dụng các phép BĐSC mô tả ở trên để đưa A về dạng chuẩn (gần đơn vị) như sau:

$$\begin{bmatrix} 1 & 0 & 0 & . \\ 0 & 1 & 0 & . \\ 0 & 0 & 1 & . \\ . & . & . & 0 \end{bmatrix}$$

Kết quả của bài toán này thể hiện trên màn hình. Lần lượt thể hiện các ma trận trung gian và ghi rõ các phép BĐSC tương ứng, nhấn một phím để chuyển sang màn hình tiếp theo.

17. Sử dụng thuật toán của bài 16 để xây dựng thủ tục kiểm tra xem ma trận vuông số cho trước có phải là khả nghịch hay không?

18. Trên một lưới N×N ô vuông được đánh các số 1 hoặc -1. Lưới trên được biến đổi theo qui tắc sau: Một ô nào đó được thay thế bởi tích của tất cả các số trong các ô kề nó (kề cạnh). Bạn hãy lập chương trình thực hiện các nước đi sao cho thu được một lưới gồm toàn các số 1. Các nước đi cần thể hiện trên màn hình, vị trí ô trung tâm của biến đổi được ghi rõ.

19. Bài toán sau mô tả thuật toán tính ma trận ngược của một ma trận số cho trước.
Giả sử cho trước ma trận số vuông A. Giả sử dãy BĐSC F1, F2, ..., Fk dùng để đưa A về dạng ma trận gần đơn vị (theo bài 16).
Khi đó dãy các phép biến đổi này khi áp dụng cho ma trận đơn vị sẽ biến đổi ma trận đơn vị E này về ma trận ngược A^{-1}.
a. Viết chương trình nhập dữ liệu là ma trận số cho trước A, sử dụng thuật toán trên để tìm ma trận ngược của A.
b. Hãy chứng minh tính đúng đắn của thuật toán trên.

20. Trên một lưới toạ độ N × N ô vuông, tại góc trái dưới với toạ độ (1, 1) có đặt một hòn bi. Mỗi bước đi của trò chơi theo qui luật sau: Nếu hòn bi ở toạ độ (i, j) mà ở các toạ độ (i+1, j) và (i, j+1) còn trống thì bỏ hòn bi này đi và đặt vào 2 toạ độ (i+1, j) và (i, j+1) hai hòn bi khác. Mục đích của trò chơi là đẩy các hòn bi càng xa gốc toạ độ càng tốt. Cụ thể: Đẩy các hòn bi khỏi tam giác vuông cân đỉnh ở gốc toạ độ và có cạnh bằng d.

Bạn hãy lập chương trình thực hiện mục đích sau ít bước đi nhất.

Bài 7. Số ngẫu nhiên. Tính xấp xỉ giá trị số

Bài tập

1. Nói chung ta không thể tìm được nghiệm chính xác của một phương trình bậc cao. Do đó trong nhiều trường hợp ta phải giải gần đúng.
Giả sử f(x) là hàm liên tục và phương trình f(x) = 0 có đúng 1 nghiệm trong đoạn [a, b], tức f(a)× f(b) < 0. Để tìm nghiệm đó ta làm như sau:

- Chia [a, b] thành 2 đoạn bởi điểm c = (a+b)/2, sau đó xét dấu f(c). Nếu f(c) cùng dấu f(a) thì đặt a:=c, nếu f(c) ngược dấu f(a), tức cùng dấu f(b) thì đặt b:=c.
- Lặp lại quá trình trên cho tới khi đạt được độ chính xác cho trước. Độ chính xác tính theo |a-b| hoặc |f(x)|.

Bạn hãy lập chương trình tìm nghiệm gần đúng nằm trong đoạn [-5, 5] của phương trình:

$$x^3 + x^2 + x + 1 = 0$$

theo cách trên với độ chính xác 0.0001. Kết quả đưa ra màn hình.

2. Sử dụng cách làm trong bài 1, tính nghiệm gần đúng trong đoạn [2.1,2.2] của phương trình:

$$x^3 - 2x^2 + x - 3 = 0$$

với độ chính xác đến 0.0001. Kết quả đưa ra màn hình.

3. Sử dụng cách làm trong bài trên, tính nghiệm gần đúng trong đoạn [0.5, 0.6] của phương trình:

$$x^2 - \sin 5x = 0$$

với độ chính xác đến 0.01. Kết quả đưa ra màn hình.

4. Giả sử f(x) là hàm khả vi và phương trình f(x) = 0 có 1 nghiệm nằm gần x_0. Để tìm nghiệm gần đúng của phương trình trên ta làm như sau:

- Đặt $x:=x_0$.
- Gán $x:=x-f(x)/f'(x)$.
- Lặp lại quá trình trên cho tới khi đạt được độ chính xác cho trước. Độ chính xác tính theo |x(bước trước) - x(bước sau)| hoặc |f(x)|.

Bạn hãy lập chương trình tìm nghiệm gần đúng nằm gần $x_0=0$ của phương trình $x^3 + x^2 + x + 1 = 0$ theo cách trên với độ chính xác 0.0001. Kết quả đưa ra màn hình.

5. Sử dụng cách làm của bài 4 hãy tìm nghiệm gần đúng gần $x_0 = 2.2$ của phương trình:

$$x^3 - 2x^2 + x - 3 = 0 \text{ với độ chính xác đến } 0.0001.$$

6. Làm tương tự bài 4 với $x_0 = 4.6$ và phương trình: tgx - x = 0 với độ chính xác 0.0001.

7. Tính số π với độ chính xác 10^{-10} theo công thức sau:

$$\frac{\pi}{4} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots$$

8. Tính số e với độ chính xác 10^{-10} theo công thức sau:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \ldots$$

9. Tính hàm số Exp(x)= e^x với độ chính xác 10^{-10} theo công thức:

$$e^x = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \ldots$$

x nhập từ bàn phím.

10. Để tính gần đúng tích phân xác định, người ta có thể dùng công thức sau, gọi là *công thức hình chữ nhật*:

$$\int_a^b f(x)dx \approx h\left[f(x_1) + f(x_2) + \ldots + f(x_n)\right]$$

trong đó h = (b-a)/n, x_i = a + ih + h/2.
Bạn hãy lập thủ tục tính gần đúng tích phân xác định của hàm f(x) từ a đến b theo công thức hình chữ nhật.

11. Để tính gần đúng tích phân xác định, người ta có thể dùng công thức sau, gọi là *công thức hình thang*:

$$\int_a^b f(x)dx \approx h\left[f(x_0)/2 + f(x_1) + f(x_2) + \ldots + f(x_n)/2\right] = I_n$$

trong đó h = (b-a)/n, x_i = a + ih.
Bạn hãy lập thủ tục tính gần đúng tích phân xác định của hàm f(x) từ a đến b theo công thức hình thang. Để đạt được độ chính xác cần thiết, người ta chọn số n_0 tuỳ ý, sau đó tính I_n với $n = n_0, 2n_0, 4n_0, \ldots$
Việc tính toán được dừng lại khi $|I_{2n} - I_n|/3 < e$ là độ chính xác cần thiết.

12. Dùng các thủ tục lập ở 2 bài trên tính gần đúng tích phân xác định của hàm y = ln(2+sinx) từ 1 đến 2. So sánh các kết quả.

Tính gần đúng các tích phân sau với độ chính xác 0.001

13. $$\int_0^6 \frac{dx}{\sqrt{1+x^4}}$$

14. $$\int_0^{\pi} \sqrt{1+\cos^2 x}\, dx$$

15. $$\int_0^1 \frac{dt}{\sqrt{(1+t^2)(4+t^2)}}$$

16. *Dãy số ngẫu nhiên.*

Trên thực tế người ta thường hay sử dụng tính chất và các tính toán dựa trên khái niệm dãy số ngẫu nhiên trong một khoảng nào đó.

Định nghĩa: *Dãy số $x_1, x_2, ..., x_n$ nằm trong khoảng (0,1) được gọi là dãy số ngẫu nhiên phân bố đều trong khoảng (0,1) hay ngắn gọn là dãy số ngẫu nhiên trong khoảng (0,1) nếu với mọi khoảng con (a,b) bên trong (0,1), xác suất để x_n nằm trong (a,b) sẽ là b-a.*

Trong Pascal đoạn chương trình sau sẽ tạo ra một dãy ngẫu nhiên phân bố đều trên khoảng (0,1).

```
i:=1;
Randomize;
Repeat
  x[i]:=random;
  i:=i+1;
until i > n;
```

Hãy viết chương trình sinh ngẫu nhiên dãy 10000 phần tử trong khoảng (0,1). Kết quả đưa ra tệp văn bản RANDOM.DAT, mỗi số một dòng, đồng thời thể hiện trên màn hình.

17. Xây dựng chương trình sinh dãy bao gồm 2 số 0, 1 ngẫu nhiên với độ dài bất kỳ.

18. Viết chương trình sinh dãy nhị phân ngẫu nhiên độ dài k cho trước.

19. Viết chương trình sinh dãy ngẫu nhiên từ các số tự nhiên 1, 2, ..., 9.

20. *Dãy số ngẫu nhiên với phân bố*

Cho trước các phần tử $a_1, a_2, ..., a_n$ và các số $p_1, p_2, ..., p_n > 0$ sao cho $p_1 + p_2 + ... + p_n = 1$.

Ta nói dãy $V_1, V_2, ...$, là *dãy ngẫu nhiên với phân bố* $\begin{bmatrix} a_1 & a_2 & ... & a_n \\ p_1 & p_2 & ... & p_n \end{bmatrix}$,

nếu thỏa mãn điều kiện sau:

- Dãy $V_1, V_2, ...$ bao gồm các phần tử có giá trị nằm trong dãy $a_1, a_2, ..., a_n$.
- Xác suất để xuất hiện phần tử a_i trong dãy này là p_i.

Thuật toán xây dựng một dãy ngẫu nhiên với phân bố cho trước như sau:

Bước 1. Chia khoảng (0,1) thành n khoảng con với độ dài lần lượt là $p_1, p_2, ..., p_n$.

Các điểm chia sẽ là p_1, $p_1 + p_2$, $p_1 + p_2 + p_3$, ...

Ta gọi các khoảng con này là $I_1, I_2, ..., I_n$.

Bước 2. Sinh ra một dãy ngẫu nhiên $X_1, X_2, ...$ trong khoảng (0,1).

Bước 3. Dãy $\{V_i\}$ được xây dựng như sau:

- Nếu X_i nằm trong khoảng I_k ta đặt $V_i = a_k$
- Nếu X_i nằm trên một điểm chia, gọi I_k là khoảng bên phải, ta đặt $V_i = a_k$.

Hãy viết chương trình sinh ra một dãy ngẫu nhiên với phân bố đều của (1, 2, 3, 5, 6).

21. Viết chương trình sinh một dãy số ngẫu nhiên với phân bố đều của (0, 1, 2, 3, ..., 9).

22. Viết chương trình sinh ngẫu nhiên dãy với phân bố sau:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

Bài 8. Xử lí văn bản

Bài tập

1. Cho một văn bản chứa trong một text file. Bạn hãy viết chương trình liệt kê các chữ cái chỉ có mặt trong văn bản đúng 1 lần theo thứ tự của bảng chữ cái.

2. Cho một văn bản chứa trong một text file. Bạn hãy viết chương trình thực hiện các công việc sau đây:
- Đếm xem văn bản trên chứa bao nhiêu dòng
- Tính xem trung bình một dòng của văn bản trên chứa bao nhiêu ký tự.

3. Cho một văn bản chứa trong một text file. Bạn hãy viết chương trình tính tỉ lệ các chữ cái có mặt trong văn bản theo thứ tự của bảng chữ cái.

4. Cho một văn bản chứa trong một text file. Bạn hãy viết chương trình tính tỉ lệ các từ có mặt trong văn bản theo thứ tự của bảng chữ cái.

5. Cho một văn bản chứa trong một text file. Bạn hãy viết chương trình tính tỉ lệ các nguyên âm có mặt trong văn bản theo thứ tự của bảng chữ cái.

6. Bạn hãy lập chương trình nhập 2 số nguyên dương a và b, sau đó thực hiện phép nhân $a \times b$ như cách nhân bằng tay thông thường. Ví dụ:

$$
\begin{array}{r}
37 \\
\times\ 45 \\
\hline
185 \\
148 \\
\hline
1665
\end{array}
$$

7. Nhập vào một xâu biểu diễn một đa thức theo dạng tương tự như sau: -8x^3+2x^2+0.7x-1.
Đa thức nhập vào là tổng đại số các đơn thức dạng ax^b, với b là số nguyên dương (không dấu), a là số thực dương (không dấu), viết dưới dạng thông thường. Chương trình phải:

- Kiểm tra tính đúng đắn của biểu thức.

- Qui gọn và sắp xếp đa thức theo chiều giảm của bậc đơn thức.

8. Bạn hãy lập chương trình nhập vào một xâu kí tự chỉ gồm các dấu mở ngoặc và đóng ngoặc đơn, sau đó kiểm tra tính đúng đắn của cách đặt các dấu ngoặc. Yêu cầu:

- Số lần mở ngoặc đúng bằng số lần đóng ngoặc.
- Dấu mở ngoặc phải đứng trước (ở bên trái) dấu đóng ngoặc tương ứng.

9. Định nghĩa "biểu thức số học" như sau:

- Một số nguyên không dấu.
- Hai biểu thức số học nối với nhau bởi một dấu phép tính (+, -, *, /), có thể đặt trong một cặp dấu ngoặc đơn.

Ý nghĩa của biểu thức và thứ tự các phép tính hiểu theo nghĩa thông thường. Yêu cầu lập chương trình nhập vào một xâu kí tự, sau đó kiểm tra xem xâu nhập vào có phải là một biểu thức số học đúng hay không, nếu đúng thì tính giá trị của biểu thức.

10. Mở rộng bài trên bằng cách định nghĩa "biểu thức số học" như sau:

- Một số thực không dấu hoặc có dấu (nếu có dấu phải đặt trong ngoặc đơn).
- Hai biểu thức số học nối với nhau bởi một dấu phép tính (+, -, * hoặc /), có thể đặt trong một cặp dấu ngoặc đơn.

Ý nghĩa của biểu thức và thứ tự các phép tính hiểu theo nghĩa thông thường. Yêu cầu lập chương trình nhập vào một xâu kí tự, sau đó kiểm tra xem xâu nhập vào có phải là một biểu thức số học đúng hay không, nếu đúng thì tính giá trị của biểu thức.

11. Định nghĩa "biểu thức đại số" như sau:

- Một số nguyên không dấu, hoặc một tên biến đặt theo chuẩn của Pascal.
- Hai biểu thức đại số nối với nhau bởi một dấu phép tính (+, -, * hoặc /), có thể đặt trong một cặp dấu ngoặc đơn.

Ý nghĩa của biểu thức và thứ tự các phép tính hiểu theo nghĩa thông thường. Yêu cầu lập chương trình nhập vào một xâu kí tự, sau đó kiểm tra xem xâu nhập vào có phải là một biểu thức đại số đúng hay không.

12. Bạn hãy lập chương trình nhập vào một xâu kí tự là một biểu thức đại số và giá trị của các biến theo mẫu:

(A1+15)/4 - 7*xyz

A1 = 2

xyz = 4

Sau đó kiểm tra tính đúng đắn và tính giá trị của biểu thức đó, các biến chưa được "nhập" giá trị có giá trị bằng 0.

13. Mở rộng bài trên bằng cách định nghĩa "biểu thức đại số" như sau:

- Một số thực không dấu hoặc có dấu (nếu có dấu phải đặt trong ngoặc đơn), hoặc một tên biến đặt theo chuẩn của Pascal.
- Hai biểu thức đại số nối với nhau bởi một dấu phép tính (+, -, * hoặc /), có thể đặt trong một cặp dấu ngoặc đơn.

Bạn hãy lập chương trình nhập vào một xâu kí tự là một biểu thức đại số và giá trị của các biến theo mẫu:

(A1+15.5)/(-4.3)-7.2*xyz

A1 = -2.1

xyz = 4

Sau đó kiểm tra tính đúng đắn và tính giá trị của biểu thức đó, các biến chưa được "nhập" giá trị có giá trị bằng 0.

14. Có một bàn cân hai đĩa và n quả cân khác nhau từng đôi một. Lần lượt đặt các quả cân lên một trong hai đĩa. Mỗi lần đặt thêm một quả cân ta ghi lại trạng thái của cân bằng chữ: L (nếu trái nặng hơn), R (nếu phải nặng hơn) và E (nếu cân cân bằng). Như vậy sau khi đặt n quả cân sẽ tương ứng với một dãy n kí tự L, R và E.

- Chứng minh rằng với mọi xâu n kí tự L, R, E như trên sẽ tồn tại một cách đặt quả cân tương ứng.
- Cho trước xâu kí tự gồm các chữ cái L, R, E. Bạn hãy lập chương trình chỉ ra cách đặt quả cân tương ứng. Xâu kí tự (có thể rất dài) chứa trong một text file, kết quả cũng được ghi ra một text file khác.

15. Một số tự nhiên gồm 2m chữ số được gọi là 'tốt' nếu tập hợp các chữ số có chỉ số chẵn và tập hợp các chữ số có chỉ số lẻ chứa cùng một số các chữ số chẵn (và lẻ). Ví dụ số 2347169824 là số tốt vì tập hợp các chữ số có chỉ số chẵn và tập hợp các chữ số có chỉ số lẻ đều chứa 3 chữ số chẵn (và 2 chữ số lẻ).

Cho trước một số có 2m+1 chữ số. Bạn hãy lập chương trình xoá đi của nó một chữ số sao cho nó trở nên số tốt. Số này được ghi trong một text file, kết quả cũng được ghi trong file đó đồng thời với dòng thông báo chữ số bị xóa là chữ số nào và ở vị trí nào.

16. Cho tệp văn bản (file kiểu TEXT) có tên DULIEU.INP chứa n dòng $S_1, ..., S_n$ với $n \geq 3$, mỗi dòng đều không trống. Nói rằng xâu P là thành phần của dòng S_i và ký hiệu là $P < S_i$, nếu P có độ dài lớn hơn 1 và có thể nhận được từ S_i bằng cách xoá một số ký tự của S_i. Nói P là thành phần chung của các dòng $S_1, S_2, ..., S_n$ nếu $P < S_i$ với mọi i = 1, 2, ..., n.

a. Hãy cho biết file đã cho có bao nhiêu dòng và tìm một thành phần chung P của tất cả các dòng S_i đã cho trong file.

b. Tìm thành phần chung Q có độ dài lớn nhất của $S_1, S_2, ..., S_n$.

Yêu cầu: Với hai câu a và b trên hãy hiển thị:

- Với mỗi i = 1,2, .., n hiện tất cả các vị trí lần lượt bị xoá trong dòng S_i.
- P, Q và độ dài của chúng.
- Nếu P và Q không tồn tại thì hãy thông báo điều đó.

c. Tìm k lớn nhất thoả mãn: $Q < S_{m1} < S_{m2} < ... < S_{mk}$, trong đó S_{mj} (j = 1,2,...,k) là những dòng trong tệp đã cho. Hiển thị k và các m_j theo trật tự nói trên.

d. Với i,j bất kỳ nhập từ bàn phím hãy tìm cách biến đổi dòng S_i thành S_j bằng cách sử dụng ít nhất có thể được các phép biến đổi, đối với từng trường hợp sau:

- Chỉ sử dụng hai loại phép toán: chèn một ký tự, xóa một ký tự.
- Chỉ sử dụng ba loại phép toán: chèn một ký tự, xoá một ký tự và thay một ký tự trong xâu bằng một ký tự khác tùy ý.

Hãy hiển thị lần lượt vị trí và phép toán đã được sử dụng, tổng số các phép biến đổi mỗi loại.

17. Xâu ký tự thuần nhất (được định nghĩa) là xâu chỉ bao gồm các chữ cái tiếng Anh. Một xâu thuần nhất có thể được viết dưới dạng thu gọn, bao gồm các nhóm ký tự có thể kèm theo số lần xuất hiện liên tiếp của nhóm đó.

Ví dụ: Xâu 'XCAABAABAABCCADADCAABAABAABCCADADY' có thể viết dưới dạng thu gọn: 'X(C(A2B)3C2(AD)2)2Y'.

Cho một file kiểu TEXT có tên 'XAU' gồm các dòng ký tự, mỗi dòng là một xâu.

a. Hãy đọc lần lượt từng xâu của file đã cho, thông báo xâu đó thuộc những dạng nào trong số 3 dạng dưới đây:

- Dạng thuần nhất.
- Dạng thu gọn của dạng thuần nhất.
- Dạng sai: xâu không phải là dạng thuần nhất và cũng không phải là dạng thu gọn.

b. Nếu xâu đọc được thuộc dạng thuần nhất, hãy biến đổi nó về dạng thu gọn có chiều dài ngắn nhất có thể được. Nếu xâu đọc được thuộc dạng thu gọn, hãy biến đổi nó trở lại dạng thuần nhất tương ứng. Đối với mỗi xâu đọc được hãy hiển thị:

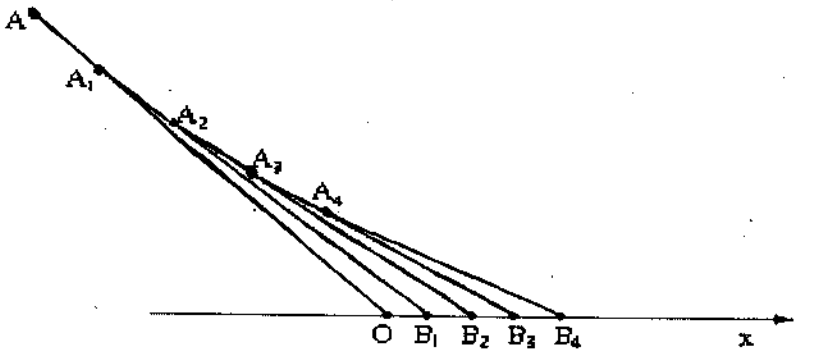
- Chính xâu đó.
- Loại xâu.
- Xâu đã được biến đổi.
- Đối với xâu sai, hãy chỉ ra một nguyên nhân dẫn đến sai.

Bài 9. Vật lí và sinh học

Bài tập

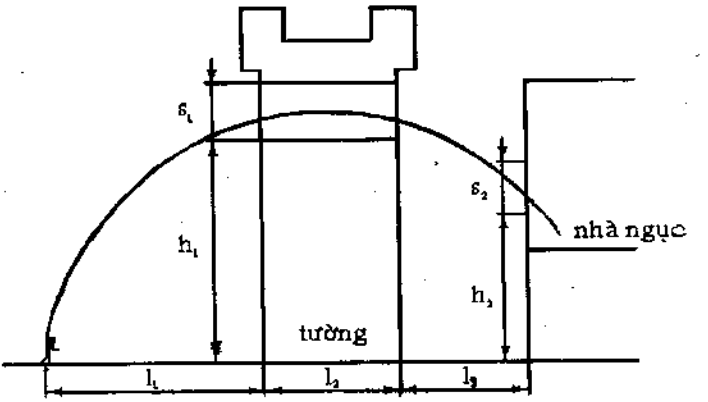
1. Hệ gồm 10 chất điểm trong không gian được cho bởi mảng số thực $A = (a_{ij})$ kích thước 4×10, trong đó mỗi cột chứa số liệu của một điểm; các hàng từ thứ nhất đến thứ ba chứa các toạ độ của điểm, còn hàng thứ tư chứa khối lượng của điểm. Hãy lập chương trình tìm trọng tâm của hệ và khoảng cách từ đó tới từng điểm của hệ. Dữ liệu cho trong tệp INPUT.DAT có khuôn dạng như mô tả ở trên. Kết quả đưa ra màn hình.

2. Một quả cầu bán kính r, khối lượng riêng $\rho < 1$ được thả xuống nước. Bạn hãy lập chương trình tính khoảng cách từ mặt nước tới tâm quả cầu. Các số r, ρ nhập từ bàn phím.



3. Một con chó đuổi theo một con mèo. Tại thời điểm đầu tiên t = 0 chó ở vị trí A, mèo ở vị trí O. Khoảng cách AO = 100m, góc xOA = 135^0. Hướng chạy của con mèo là hướng Ox, còn con chó luôn hướng theo con mèo. Vận tốc của chó là 10m/s, của mèo là 5m/s. Để vẽ được đường đi của con chó, ta có thể làm theo cách gần đúng như sau: Chia thời gian thành các quãng đều nhau T. Trong quãng thời gian đầu tiên $t \approx [0,T]$ con chó chạy theo hướng AO được một đoạn AA_1, con mèo chạy được một đoạn OB_1, trong quãng thời gian tiếp theo $t \approx [T,2T]$ con chó chạy theo hướng A_1B_1 được một đoạn A_1A_2, con mèo chạy được một đoạn B_1B_2... Tất nhiên T càng nhỏ thì quĩ đạo càng chính xác:
Bạn hãy lập chương trình vẽ quĩ đạo của con chó trên màn hình.

4. Bạn hãy viết chương trình giải bài toán "Bức thư của Robin Hud gửi tù nhân của pháo đài Notingham" như hình vẽ sau đây:



Biết các độ dài l_1, l_2, l_3 và các chiều cao s_1, h_1, s_2, h_2. Chiều cao của Robin Hud là 1.8m. Hỏi muốn ném được bức thư vào nhà ngục thì anh phải ném với vận tốc đầu và góc ném bằng bao nhiêu? Gia tốc g lấy bằng $10m/s^2$, bỏ qua sức cản của không khí. Toàn bộ dữ liệu ban đầu nhập từ bàn phím.

5. Nhóm thứ nhất gồm k người bị mắc một bệnh truyền nhiễm. Điều tra một nhóm thứ hai gồm *l* người xem có tiếp xúc với nhóm thứ nhất hay không. Kết quả được ghi vào một ma trận nhị phân $A = (a_{ij})$ cấp $l \times k$, với $a_{ij} = 1$ khi và chỉ khi người thứ i trong nhóm thứ hai tiếp xúc với người thứ j trong nhóm thứ nhất. Tiếp tục điều tra một nhóm thứ ba gồm m người xem có tiếp xúc với nhóm thứ hai hay không. Kết quả được ghi vào một ma trận nhị phân $B = (b_{ij})$ cấp $k \times m$, với $b_{ij} = 1$ khi và chỉ khi người thứ i trong nhóm thứ ba tiếp xúc với người thứ j trong nhóm thứ hai. Bạn hãy lập chương trình tạo ma trận $C = (c_{ij})$ thể hiện sự tiếp xúc gián tiếp của nhóm thứ ba với nhóm thứ nhất, biết $c_{ij} = 1$ khi và chỉ khi người thứ i trong nhóm thứ 3 tiếp xúc gián tiếp với người thứ j ở nhóm thứ nhất (thông qua một người nào đó ở nhóm thứ hai). Dữ liệu của các ma trận A và B chứa trong các tệp A.TXT và B.TXT. Dữ liệu đầu ra của ma trận C cho trong tệp C.TXT.

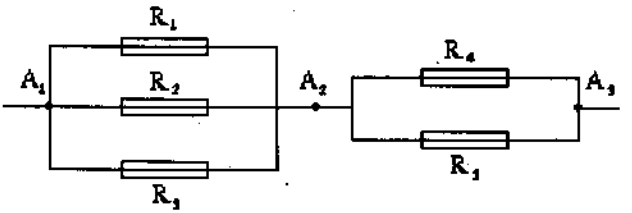
6. Sau khi tiêm một loại thuốc n giờ, phản ứng của cơ thể được đo bởi một chỉ số r_n mà theo thực nghiệm xác định được công thức:

$r_n = \alpha\, r_{n-1} + 0.4^n$, với $r_0 = 1$, α là một số dương nhỏ hơn 1 đặc trưng cho từng loại thuốc. Hãy lập chương trình để xác định xem:

a. Sau mấy giờ thì phản ứng của cơ thể với thuốc là mạnh nhất?
b. Sau mấy giờ thì phản ứng của cơ thể với thuốc giảm đi một nửa so với phản ứng ngay sau khi tiêm?

α được nhập từ bàn phím.

7. Một mạch điện một chiều có thể được mô tả như một đa đồ thị, trong đó mỗi đỉnh là một đoạn dây nối, còn mỗi cạnh là một điện trở. Ví dụ mạch điện sau:



với $R_1 = 1\Omega$, $R_2 = 2\Omega$, $R_3 = 3\Omega$, $R_4 = 4\Omega$, $R_5 = 5\Omega$
có thể mô tả như sau:

3
1 2 1
1 2 2
1 2 3
2 3 4
2 3 5

Trong đó dòng đầu tiên là số lượng đỉnh, các dòng tiếp theo mỗi dòng ứng với một cạnh gồm 3 số: 2 số đầu là chỉ số của cạnh, số thứ 3 là độ dài cạnh (điện trở R). Ví dụ dòng 2 3 5 là điện trở R_5 (nối đỉnh A_2 và A_3 và có điện trở là 5Ω).

Cho text file f với định dạng như trên. Bạn hãy lập chương trình tính điện trở chung toàn mạch từ A_1 đến A_n.

8. Trò chơi "Life":

Trên một lưới ô vuông vô hạn có một số vị trí được đánh dấu. Ta gọi chúng là Fish (con cá). Từ vị trí ban đầu bằng các luật di truyền các con Fish này sẽ bị "biến đổi" sang thế hệ sau. Sau đây là mô tả luật "Di truyền":

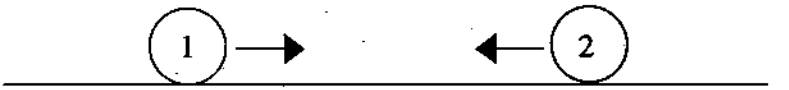
Luật gồm 3 tiêu chuẩn: Sinh ra, Chết đi, Sống sót.

- **Sinh ra:** Fish được sinh ra từ một ô trống nếu ở thế hệ trước ô này có đúng 3 Fish kề bên cạnh (hai ô gọi là kề nếu chúng chung cạnh hoặc đỉnh).
- **Chết đi:** Fish bị chết đi tại một vị trí nếu nó có nhiều hơn 3 hoặc ít hơn 2 Fish kề bên cạnh.
- **Sống sót:** Fish được sống sót qua thế hệ sau nếu nó có hai hoặc 3 Fish kề bên cạnh.

a. Lập chương trình mô tả quá trình sống trên.

b. Tồn tại hay không một con Fish mà nó càng ngày càng "lớn" lên đến vô tận?

9. Viết chương trình mô phỏng chuyển động của hai quả cầu giống hệt nhau trên một đường thẳng (xem sơ đồ sau):



Vị trí ban đầu của hai quả cầu là X_1, X_2. Vận tốc ban đầu là V_1, V_2. Các dữ liệu được nhập từ bàn phím.

Chương trình mô phỏng khoảng thời gian tối thiểu để hai quả cầu đó có thể va chạm nhau.

10. Tập hữu hạn các số tự nhiên gọi là "Hệ bất trọng" nếu từ tập này không thể trích ra hai dãy con rời nhau với tổng số bằng nhau.

Ví dụ: 1, 3, 7, 20 lập thành hệ bất trọng.

a. Chứng minh rằng dãy 1, 2, 4, ..., 2k-1 là bất trọng.

b. Giả sử $A_1 < A_2 < ... < A_k$ - hệ bất trọng. Chứng minh rằng:

$$\frac{1}{A_1}+\frac{1}{A_2}+...+\frac{1}{A_k} < 2.5$$

c. Cho k số tự nhiên $A_1 < A_2 < ... < A_k$ thoả mãn tính chất: Mọi tổng của hai hệ con cùng độ dài đều khác nhau.

Ta chọn số A thoả mãn:

Nếu k = 2m (k chẵn) thì

$$(A-A_{2m}) + ... + (A-A_m) > (A - A_{m-1}) + ... + (A-A_1) + A$$

Nếu k = 2m +1 (k lẻ) thì

$$(A-A_{2m+1}) + ... + (A-A_{m+1}) > (A-A_{m-1}) +... + (A-A_1) + A$$

Chứng minh rằng khi đó dãy A, $A-A_1$, $A-A_2$, ..., $A-A_k$ sẽ lập thành một hệ bất trọng.

d. Dãy $A_1 < A_2 < .... < A_n < ....$ được xây dựng như sau:

$$A_1 = 0, A_2 = 1.$$

$$A_{2n+1} = (A_{n+1} + A_{n+2} + ... + A_{2n}) - (A_1 + A_2 + ... + A_{n-1}) + 1$$

$$A_{2n+2} = (A_{n+1} + A_{n+2} + ... + A_{2n+1}) - (A_1 + A_2 + ... + A_n) + 1$$

sẽ thoả mãn tính chất được nêu trong c.

Do đó ta có thuật toán có thể xây dựng hệ bất trọng độ dài bất kỳ.

e. Xây dựng các dãy X_n, Y_n, Z_n theo bảng sau:

n	0	1	2	3	4	5	6	7	8	9	10
X_n	0	1	2	2	3	3	3	4	4	4	4
Y_n	0	0	0	1	1	2	3	3	4	5	6
Z_n	0	1	2	4	7	13	24	44	84	161	309

Dãy trên được xây dựng theo qui tắc sau:

$$Y_n = n - X_n$$

$$Z_0 = 0, Z_1 = 1, Z_{n+1} = 2Z_n - Y_n$$

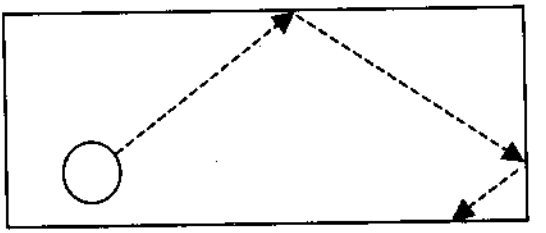
Chứng minh rằng dãy:

$$a_0 = Z_n - Z_0, a_1 = Z_n - Z_1, ..., a_{n-1} = Z_n - Z_{n-1}$$

sẽ lập thành hệ bất trọng.

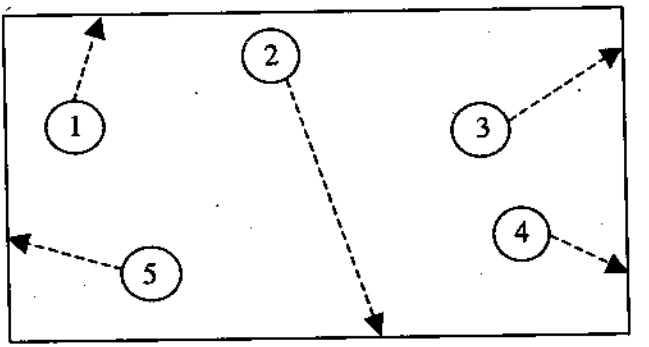
11. Viết chương trình mô phỏng chuyển động của n quả cầu giống hệt nhau trên một đường thẳng. Vị trí ban đầu của các quả cầu này là X_1, X_2, ..., X_n. Vận tốc ban đầu của các quả cầu này là V_1, V_2, ..., V_n. Dữ liệu cho trong tệp INPUT.DAT có dạng: dòng đầu tiên ghi số n, dòng thứ hai ghi n số X_1, X_2, ..., X_n, dòng thứ ba ghi n số V_1, V_2, ..., V_n. Chương trình mô phỏng khoảng thời gian tối thiểu để các quả cầu này có thể va chạm nhau.

12. Viết chương trình mô phỏng chuyển động của một quả cầu trong một không gian hình chữ nhật.



Vị trí ban đầu của quả cầu cho bởi toạ độ (x_0, y_0). Kích thước hình chữ nhật cho bởi toạ độ (x_1, y_1, x_2, y_2). Vận tốc ban đầu của quả cầu cho bởi vectơ (v_x,v_y). Dữ liệu nhập từ bàn phím.
Chú ý: Chương trình phải mô phỏng được phản xạ của quả cầu khi va chạm vào biên của hình chữ nhật (xem hình vẽ).

13. Viết chương trình mô phỏng chuyển động của n quả cầu giống nhau trong một hình chữ nhật cho trước. Các quả cầu được coi như có thể tích bằng 0. Thông số ban đầu vị trí các quả cầu (x_1,y_1), (x_2,y_2), ..., (x_n,y_n). Các vectơ vận tốc ban đầu của các quả cầu là v_1, v_2, ..., v_n. Dữ liệu được nhập từ một tệp INPUT.DAT có dạng sau:



- Dòng đầu tiên ghi toạ độ hình chữ nhật: tọa độ các đỉnh trái dưới và phải trên.
- Dòng thứ hai ghi số n
- n dòng tiếp theo ghi lần lượt tọa độ của n quả cầu, mỗi dòng 2 số cách nhau bởi dấu cách.
- n dòng tiếp theo ghi lần lượt n vectơ vận tốc ban đầu của các quả cầu trên, mỗi vectơ bao gồm 2 số cách nhau bởi dấu cách.

14. Để mô tả quá trình phát triển của một hệ sinh thái người ta phân loại các sinh vật sống theo các nhóm tuổi như sau:
Nhóm thứ nhất bao gồm tất cả các loài có tuổi nhỏ hơn 1 năm, nhóm thứ hai bao gồm các loài không nằm trong nhóm thứ nhất và có tuổi nhỏ hơn 2 năm, ... Giả sử P_n là số các loài có trong nhóm thứ n. Như vậy hệ sinh thái tại thời điểm hiện thời được mô tả theo phân loại (P_1, P_2, P_3, ..., P_n, ...). Để đơn giản ta giả sử rằng mỗi loài đều sống đúng N năm (!). Do vậy phân loại của chúng ta sẽ có dạng (P_1, P_2, P_3, ..., P_N). Giả sử tỷ lệ sinh của các nhóm theo hàng năm lần lượt là b_1, b_2, ..., b_N. Do đó phân loại sinh thái cho năm tiếp theo sẽ là ($b_1P_1 + b_2P_2 + ... + b_NP_N$, P_1, ..., P_{N-1}). Giả sử các giá trị N, P_1, P_2, ... P_N, b_1, b_2, ..., b_N đã cho sẵn. Viết chương trình nhập số tự nhiên k và tính phân loại sinh thái cho k năm tiếp theo.

15. Một chất thể phẳng được cho bởi n tọa độ điểm M_1, M_2, ..., M_n với trọng lượng tương ứng m_1, m_2, ..., m_n. Input cho bởi file text có dạng: n dòng, mỗi dòng bao gồm tọa độ của điểm M_i và trọng lượng m_i tương ứng. Hãy lập chương trình tìm ra tọa độ trọng tâm của chất thể trên. Kết quả đưa ra màn hình.

16. Máy đọc thẻ
Có 3 máy đọc và in thẻ, mỗi thẻ được in ra sẽ chứa một cặp có thứ tự 2 số tự nhiên (a,b):

- Máy thứ nhất nếu đọc thẻ (a,b) sẽ in ra thẻ (a+1,b+1).
- Máy thứ hai nếu đọc thẻ (a,b) mà a và b đều chẵn sẽ in ra thẻ (a/2,b/2).
- Máy thứ ba nếu đọc thẻ (a,b) và thẻ (b,c) sẽ in ra thẻ (a,c).

Mỗi thẻ đều có thể đọc lại nhiều lần.

a. Giả sử ban đầu ta chỉ có duy nhất một thẻ (5,19). Hỏi có thể dùng 3 máy trên để in ra thẻ (1,50), thẻ (1,100) được hay không?
b. Giả sử ban đầu ta chỉ có duy nhất một thẻ (a,b) và ta muốn in ra thẻ (1,n). Hỏi với n bằng bao nhiêu thì có thể dùng 3 máy trên để in ra thẻ đó? Nếu in được thì bạn hãy lập chương trình chỉ ra thứ tự sử dụng các máy cùng các thẻ tương ứng.

Bài 10. Âm nhạc và hội hoạ

Bài tập

1. Dãy tần số (cao độ) $\{a_i\}$ $(1 \le i \le 96)$ của các âm (nốt) dùng trong âm nhạc tạo thành một cấp số nhân với công bội q tính bằng công thức:

$$q = \sqrt[12]{2} = \mathrm{Exp}\left[\frac{\ln(2)}{12}\right]$$

Cho biết cao độ của nốt thứ 37 là 261Hz. Hãy lập chương trình tính cao độ của các nốt khác trong dãy và lần lượt phát ra các nốt đó. Cao độ của các nốt được làm tròn tới đơn vị.

2. Trong dãy nốt nhạc ở bài trên, cứ 12 nốt liên tiếp lập thành một quãng tám. Các nốt từ 1 đến 12 lập thành quãng tám thứ 0, các nốt từ 13 đến 24 lập thành quãng tám thứ 1... Trong mỗi một trong 8 quãng tám trên, các nốt được đặt tên và kí hiệu lần lượt là: đô (C), đô thăng (CF, còn gọi là rê giáng), rê (D), rê thăng (DF, còn gọi là mi giáng), mi (E), fa (F), fa thăng (FF, còn gọi là sol giáng), sol (G), sol thăng (GF, còn gọi là la giáng), la (A), la thăng (AF, còn gọi là si giáng), si(B). Hỏi:

- Bạn có nhận xét gì về cao độ của các nốt cùng tên trong các quãng tám khác nhau?
- Các nốt thứ 37, 40 là nốt nào? Chúng nằm ở quãng tám nào?

3. Một dãy con gồm các nốt thứ i, i+2, i+4, i+5, i+7, i+9, i+11 (và các nốt cùng tên khác) lập nên một gam trưởng, tên của gam lấy theo tên của nốt thứ i. Ví dụ gam đô trưởng gồm các nốt đô (thứ 1), rê (thứ 3), mi (thứ 5), fa (thứ 6), sol (thứ 8), la (thứ 10), si (thứ 12); gam rê trưởng gồm các nốt thứ rê (3), mi (5), fa thăng (7), sol (8), la (10), si (12), đô thăng (14). Bạn hãy:

- Liệt kê lần lượt tên 7 nốt của gam mi trưởng, của gam sol trưởng.
- Lập chương trình phát ra lần lượt 8 nốt liên tiếp của một gam trưởng theo thứ tự xuôi và ngược, nốt đầu tiên lấy trong quãng tám thứ 3, tên của gam được nhập từ bàn phím.

4. Một hợp âm trưởng gồm 3 nốt: thứ nhất, thứ ba và thứ năm của một gam trưởng. Ví dụ hợp âm đô trưởng gồm các nốt đô - mi - sol, hợp âm rê trưởng gồm các nốt rê - fa thăng - la. Bạn hãy lập chương trình phát

ra một hợp âm trưởng với 3 nốt phát ra lần lượt, tên hợp âm nhập từ bàn phím.

5. Một dãy con gồm các nốt thứ i, i+2, i+4, i+5, i+7, i+9, i+11 (và các nốt cùng tên khác) lập nên một gam thứ, tên của gam lấy theo tên của nốt thứ i. Ví dụ gam đô thứ gồm các nốt đô (thứ 1), rê (thứ 3), mi (thứ 4), fa (thứ 6), sol (thứ 8), la (thứ 10), si (thứ 12); gam rê trưởng gồm các nốt rê (3), mi (5), fa thăng (7), sol (9), la (10), si (12), đô thăng (14). Bạn hãy:

- Liệt kê lần lượt tên 7 nốt của gam mi thứ, của gam sol thứ.
- Lập chương trình phát ra lần lượt 8 nốt liên tiếp của một gam thứ theo thứ tự xuôi và ngược, nốt đầu tiên lấy trong quãng tám thứ 3, tên của gam được nhập từ bàn phím.

6. Một hợp âm thứ gồm 3 nốt: thứ nhất, thứ ba và thứ năm của một gam thứ. Ví dụ hợp âm mi thứ gồm các nốt mi - sol - si, hợp âm rê thứ gồm các nốt rê - fa - la. Bạn hãy lập chương trình phát ra một hợp âm thứ với 3 nốt phát ra lần lượt, tên hợp âm nhập từ bàn phím.

7. Lập thủ tục:

Procedure Play(Octave: integer; Note: String; Duration: integer);

để chơi nốt Note trong quãng tám thứ Octave trong thời gian Duration miligiây.

Dùng thủ tục đó lập chương trình phát ra giai điệu bài hát "Cho con" của tác giả Phạm Trọng Cầu. Bản nhạc được ghi trong text file theo mẫu:

5 C D
5 C MD
5 D MD
4 G D
5 M D
5 D T
...

mỗi nốt được ghi trên một dòng theo thứ tự: quãng tám, tên nốt (kí hiệu) và kí hiệu trường độ phân cách nhau bởi một số dấu trống. Chữ T kí hiệu nốt trắng (Duration = 1000), chữ D- nốt đen (Duration = 500), chữ MD- móc đơn (Duration = 125)...

8. Lập chương trình liệt kê danh sách một số bài hát để người chơi lựa chọn, sau đó phát ra giai điệu của bài hát được lựa chọn.

9. Lập chương trình trò chơi phát ra giai điệu của một số bài hát quen thuộc, người chơi phải đoán xem đó là bài hát nào và của tác giả nào. Danh sách các bài hát và các tác giả được liệt kê để người chơi lựa chọn.

10. Lập chương trình trò chơi phát ra một làn điệu dân ca quen thuộc, người chơi phải đoán xem đó là dân ca của vùng nào. Danh sách các vùng được liệt kê để người chơi lựa chọn. Xong làn điệu này lại tiếp tục phát làn điệu khác.

11. Lập chương trình trò chơi phát ra giai điệu của 3 bài hát quen thuộc, sau đó phát lại một giai điệu theo thứ tự ngược lại (từ nốt nhạc cuối đến nốt nhạc đầu). Người chơi phải đoán xem đó là bài hát nào. Danh sách các bài hát được liệt kê để người chơi lựa chọn.

12. Lập chương trình trò chơi phát ra giai điệu được gõ từ bàn phím như gõ phím đàn piano. Chỉ sử dụng hàng phím giữa như hàng phím đàn.

13. Lập chương trình trò chơi nhập một đoạn nhạc từ bàn phím, sau đó phát lại giai điệu đã được nhập.

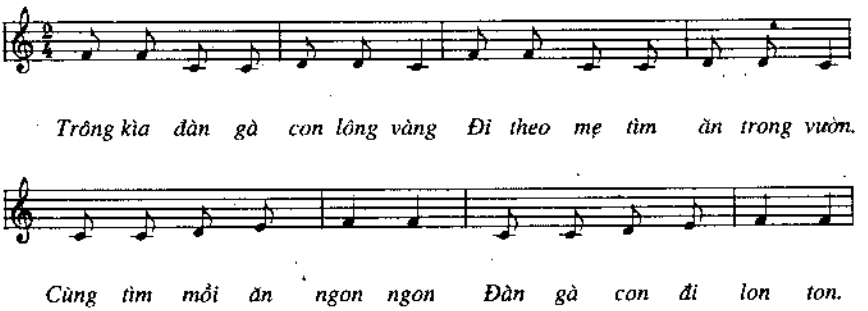
14. Lập chương trình trò chơi nhập một đoạn nhạc bằng chuột, sau đó phát lại giai điệu đã được nhập.

15. Trong chuẩn MIDI, các nốt nhạc được ghi như sau:

x ON y: bật nốt y tại thời điểm x.

x OF y: tắt nốt y tại thời điểm x.

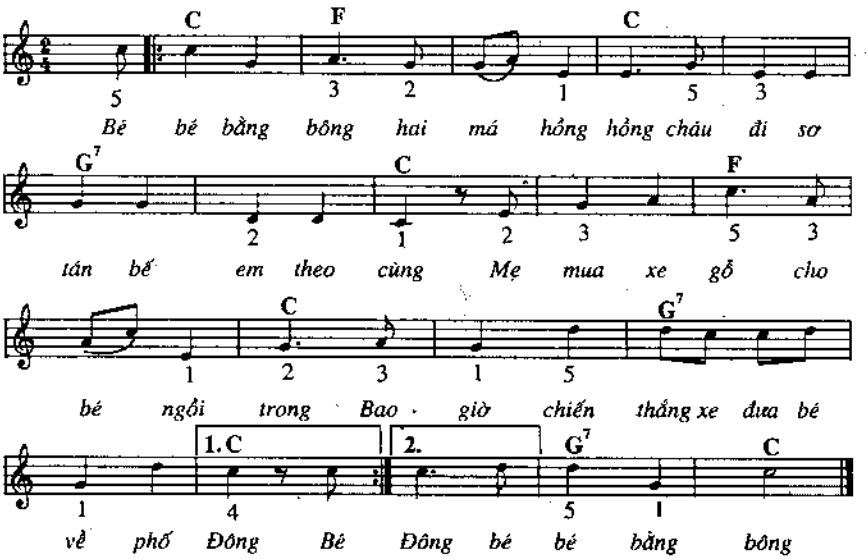
Ví dụ đoạn nhạc:



sẽ được dịch ra chuẩn MIDI như sau:

0 ON 54	100 OF 54	200 OF 53
10 OF 54	100 ON 49	200 ON 54
10 ON 54	110 OF 49	220 OF 54
20 OF 54	110 ON 49	220 ON 54
20 ON 49	120 OF 49	240 OF 54
30 OF 49	120 ON 54	240 ON 49
30 ON 49	130 OF 54	250 OF 49
40 OF 49	130 ON 54	250 ON 49
40 ON 54	140 OF 54	260 OF 49
50 OF 54	140 ON 49	260 ON 51
50 ON 54	160 OF 49	270 OF 51
60 OF 54	160 ON 49	270 ON 53
60 ON 49	170 OF 49	280 OF 53
80 OF 49	170 ON 49	280 ON 54
80 ON 54	180 OF 49	300 OF 54
90 OF 54	180 ON 51	300 ON 54
90 ON 54	190 OF 51	320 OF 54
	190 ON 53	

Hãy dịch đoạn nhạc sau sang chuẩn MIDI:



16. Lập chương trình nhập một giai điệu từ một text file ghi theo chuẩn MIDI, sau đó phát lại giai điệu đã được nhập.

17. Mô phỏng tiếng chuông điện thoại.

18. Lập chương trình nhập một giai điệu gõ từ bàn phím như gõ phím đàn piano, chỉ sử dụng hàng phím giữa như hàng phím đàn, sau đó ghi vào một text file theo chuẩn MIDI.

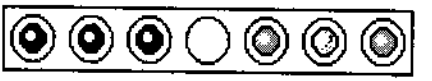
19. Mô phỏng tiếng còi hú.

Bài 11. Trò chơi

Bài tập

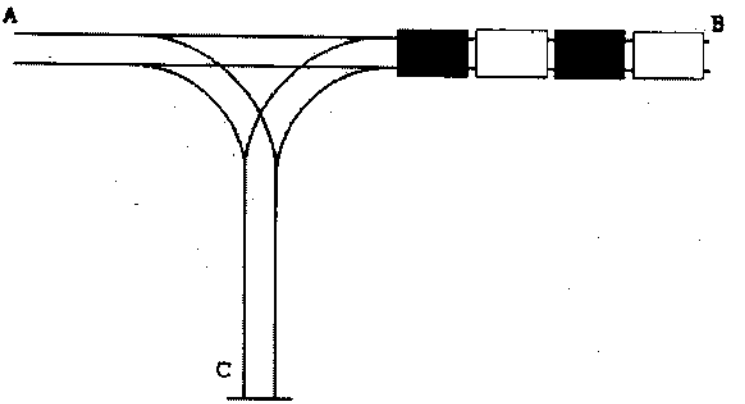
Các trò chơi từ 1 đến 10 dành cho một người chơi trên máy. Các trò chơi trong các bài từ 11 đến 22 dành cho hai người chơi, bạn hãy lập chương trình để máy chơi với người:

1. Hiển thị trên màn hình một bàn cờ 7 ô, trên đó có 6 quân cờ: 3 đen, 3 trắng như hình sau:

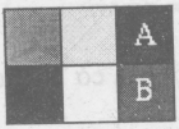


Mỗi quân cờ có thể đi vào ô trống bên cạnh hoặc vào ô trống nằm ngay sau quân bên cạnh nó. Đi một quân bằng cách kích chuột vào quân đó. Mục tiêu của trò chơi là đổi chỗ các quân trắng và các quân đen.

2. Một đường tàu có một ngách cụt và một số toa tàu màu đen và màu trắng sắp xếp một cách tuỳ ý. Số toa màu đen đúng bằng số toa màu trắng. Ngách cụt đủ chỗ để xếp tất cả các toa tàu. Mỗi toa tàu có thể di chuyển vào ngách cụt hay dọc theo đường tàu bằng cách kích chuột vào toa và vào nơi đến (A, B, C). Mục tiêu của trò chơi là đổi chỗ các toa tàu sao cho các toa màu đen và màu trắng đứng xen kẽ nhau.



3. Một bảng hình chữ nhật có kích thước 2 × 3 = 6 ô vuông, trong đó xếp 5 hình vuông màu sắc khác nhau, chỉ có 1 ô trống. Mỗi hình vuông đứng cạnh ô trống có thể trượt sang ô trống bằng cách kích chuột vào nó. Mục tiêu của trò chơi là đổi chỗ các hình vuông màu xanh (A) và màu đỏ (B), các hình vuông khác giữ nguyên vị trí.

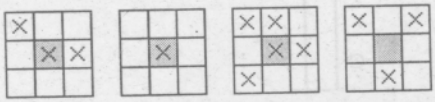


4. Một bảng hình vuông có kích thước 4×4 = 16 ô vuông, trong đó xếp 16 hình vuông chứa 16 chữ cái từ A đến P. Có thể kích chuột vào tâm mỗi hình vuông nhỏ cỡ 2×2 để quay hình vuông đó đi một góc 90^0 theo chiều kim đồng hồ, hoặc kích chuột vào giữa 2 ô vuông nhỏ để đổi chỗ chúng. Mục tiêu của trò chơi là đổi chỗ các hình vuông sao cho các chữ cái được sắp xếp theo thứ tự anphabet từ trái sang phải, từ trên xuống dưới. Trò chơi càng khó nếu thao tác thứ hai (kích chuột vào giữa 2 ô vuông nhỏ để đổi chỗ chúng) được phép sử dụng càng ít.

5. Một bảng hình vuông có kích thước 4× 4 = 16 ô vuông, trong đó xếp 15 hình vuông chứa 15 số từ 1 đến 15, chỉ còn 1 ô trống. Mỗi hình vuông đứng cạnh ô trống có thể trượt sang ô trống bằng cách kích chuột vào nó. Mục tiêu của trò chơi là đổi chỗ các hình vuông sao cho các số được sắp xếp theo thứ tự tăng dần từ trái sang phải, từ trên xuống dưới.

6. Mô phỏng cuộc sống của các tế bào sống qua các thế hệ theo qui luật sau:

- Tế bào vẫn sống nếu và chỉ nếu có 2 hoặc 3 tế bào khác trong 8 ô lân cận (a.).
- Nếu chỉ có 1 tế bào trong 8 ô lân cận hoặc không có tế bào nào thì nó sẽ chết (b.).
- Nếu có từ 4 tế bào trở lên trong 8 ô lân cận thì nó cũng sẽ chết (c.).



a. b. c. d.

• Trong một ô trống mà ở lân cận nó có đúng 3 tế bào thì sẽ sản sinh một tế bào mới ở thế hệ sau (d.).

7. Ba con ngựa thể hiện bằng 3 hình tròn nhỏ khác màu chạy đua trên một đường đua hình tròn hoặc elíp. Tốc độ của mỗi con trên từng chặng đường được chọn ngẫu nhiên. Bạn hãy đặt cược vào một trong 3 con trước khi cuộc đua bắt đầu và bạn sẽ thắng cuộc nếu như nó về đích trước.

8. Một hình tròn nhỏ tượng trưng cho quả mìn (mà người chơi không nhìn thấy được) được đặt trên màn hình. Con trỏ chuột là đầu dò. Khi đầu dò tiến tới gần quả mìn thì tần số âm thanh của máy dò sẽ tăng lên và ngược lại, khi đầu dò đi xa dần quả mìn thì tần số âm thanh của máy dò sẽ giảm xuống. Mục tiêu của trò chơi là dò được vị trí quả mìn và đánh dấu nó bằng cách kích chuột.

9. Một khẩu pháo được đặt ở cạnh dưới của màn hình và có thể di chuyển dọc theo cạnh bằng cách ấn các phím mũi tên → và ←, việc bắn được thực hiện bằng ấn phím Space (phím gõ dấu cách). Các máy bay địch bay ở phần trên màn hình với vận tốc và phương hướng khác nhau. Mục tiêu của trò chơi là bắn hạ hết máy bay địch.

10. Một hình vuông 10 ×10 ô vuông màu xanh tượng trưng cho biển, trên đó có 10 chiếc tàu ngầm của địch mà người chơi không nhìn thấy được: 4 chiếc có kích thước 1 ô vuông, 3 chiếc có kích thước 2 ô vuông, 2 chiếc có kích thước 3 ô vuông và 1 chiếc có kích thước 4 ô vuông. Vị trí của tàu được lựa chọn ngẫu nhiên. Người chơi sẽ "bắn" vào các ô vuông bằng cách kích chuột vào đó. Nếu trúng tàu thì ô vuông sẽ được tô màu đỏ, ngược lại sẽ tô màu đen. Tàu sẽ bị tiêu diệt nếu tất cả các ô của nó bị bắn trúng, khi đó nó sẽ đổi sang màu vàng. Mục tiêu của trò chơi là tiêu diệt tất cả các tàu.

11. Người chơi tự nghĩ ra một số nguyên trong khoảng từ 0 đến 1000, sau đó phải trả lời 10 câu hỏi của máy chỉ bằng hai từ đúng (T) hoặc sai (F). Máy phải tìm ra đó là số nào.

12. Bạn hãy viết một chương trình chơi trò cờ ca-rô.

13. Trên mặt đất có 2 đống sỏi. Hai người chơi trò chơi sau: Mỗi người đến lượt mình đi có quyền vứt đi một đống sỏi và chia đống sỏi còn lại làm 2 đống con (với điều kiện đống sỏi này có nhiều hơn 1 hòn sỏi). Người thắng cuộc là người đi được nước cuối cùng.

14. Bạn hãy viết một chương trình chơi trò cờ vua, trong đó máy chơi quân trắng với:

- Vua và Hậu trắng chống lại Vua đen.
- Vua và Xe trắng chống lại Vua đen.

15. Có 100 que diêm. Hai đấu thủ lần lượt bốc các que diêm theo qui tắc: mỗi lần phải bốc ít nhất là 1 que và nhiều nhất là 10 que. Người thắng cuộc là người bốc được que diêm cuối cùng.

16. Có 100 que diêm. Hai đấu thủ lần lượt bốc các que diêm theo qui tắc: mỗi lần phải bốc ít nhất là 1 que và nhiều nhất là 6 que. Người thắng cuộc là người bốc được que diêm cuối cùng.

17. Chọn thời gian: Vị trí thời gian cho bởi Ngày-Tháng, ví dụ 25-2. Hai người chơi. Người thứ nhất chọn một ngày bất kì, người thứ hai chọn tiếp cặp Ngày-Tháng sao cho Ngày>Ngày cũ hoặc Tháng>Tháng cũ hoặc là cả hai Ngày>Ngày cũ và Tháng>Tháng cũ. Người chọn được 31-12 đầu tiên sẽ là người thắng cuộc.

18. Có 24 quân cờ 4 màu: Xanh, Đỏ, Vàng, Tím. Mỗi loại 6 con có đánh số 1, 2, 3, 4, 5, 6. Hai người chơi, mỗi người được quyền lấy ra đúng 1 quân cờ và cộng số điểm vào số điểm của các quân cờ đã được lấy ra. Ai đạt được đúng 50 điểm sẽ thắng, ai đến lượt đi mà bắt buộc có nhiều hơn 50 điểm sẽ thua.

19. Trò chơi thành phố Nim: Có k đống sỏi cho trước. Hai người chơi, mỗi người có quyền lấy ra từ 1 đống sỏi số sỏi ≥ 1 bất kì. Người đi nước cuối cùng là người thắng cuộc.

20. Có n hòn sỏi trên mặt đất. Hai người chơi, mỗi người chơi được phép lấy đi một số sỏi lớn hơn 0 và nhỏ hơn hoặc bằng hai lần số sỏi người kia vừa bốc. Người đi nước cuối cùng là người thắng cuộc.

21. Trò chơi Keylesa: Có n hòn sỏi được xếp thành một hàng. Hai người chơi, mỗi người chơi được phép lấy đi từ dãy 1 hoặc 2 hòn sỏi liền nhau. Người đi nước cuối cùng là người thắng cuộc.

22. Trên mặt phẳng có 6 điểm là 6 đỉnh của một lục giác đều. Hai người chơi, mỗi người chơi được phép nối 2 đỉnh chưa được nối từ trước. Người thua cuộc là người đầu tiên phải nối thành một tam giác bởi các đường nối của mình.

Bài 12. Các thuật toán hình học

Bài tập

1. Cho 4 số thực dương a, b, c, d. Hãy kiểm tra xem có thể dựng được một tứ giác với độ dài các cạnh liền nhau là a, b, c, d hay không?.

2. Cho trước 4 số thực dương a, b, c, d. Hãy kiểm tra xem hình chữ nhật với cạnh a, b có thể chứa được toàn bộ hình chữ nhật với cạnh c, d hay không.

3. Cho 5 số thực dương a, b, c, d, e. Hãy kiểm tra xem chúng có tạo thành 5 cạnh liền nhau của một ngũ giác được không?

4. Cho hai tam giác ABC và $A_1B_1C_1$ với các cạnh a, b, c và a_1, b_1, c_1 trên mặt phẳng. Hãy kiểm tra xem tam giác ABC có thể chứa được tam giác $A_1B_1C_1$ hay không?

Trong các bài 5, 6, dữ liệu của n đường thẳng cho bởi n bộ ba số a, b, c của đường thẳng có phương trình ax + by + c = 0 được nhập từ tệp văn bản có dạng: dòng đầu tiên ghi số n, n dòng tiếp theo ghi lần lượt các bộ số a, b, c cách nhau bởi dấu cách.

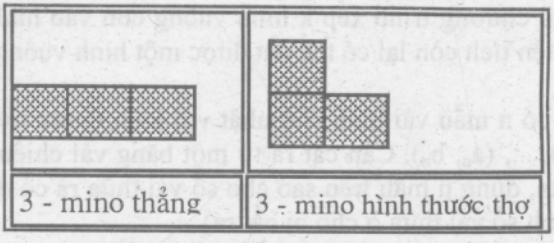
5. Cho n đường thẳng trên mặt phẳng. Hãy tính số giao điểm của n đường thẳng này.

6. Cho n đường thẳng trên mặt phẳng. Hãy tính số miền mặt phẳng được chia bởi các đường thẳng này.

7. *N-mino*: n-mino là hình thu được từ n hình vuông 1×1 ghép lại (cạnh kề cạnh). Hai n-mino được gọi là đồng nhất nếu chúng có thể đặt chồng khít lên nhau.
Bạn hãy lập chương trình tính và vẽ ra tất cả các n-mino. Số n nhập từ bàn phím.

Chú ý: Gọi M_n là số các n-mino khác nhau thì $M_1=1$, $M_2=1$, $M_3=2$, $M_4=5$, $M_5=12$, $M_6=35$,...

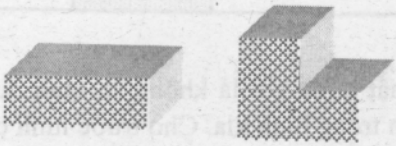
Ví dụ: Với n=3:



8. N-mino trong không gian là hình thu được từ n khối lập phương $1\times 1\times 1$ ghép lại.

Bạn hãy lập chương trình tính và vẽ ra tất cả các n-mino. Số n nhập từ bàn phím.

Ví dụ: Với n =3:



Chú ý: $M_1=1$, $M_2=1$, $M_3=2$, $M_4=7$, ...

9. Cho hình chữ nhật $M\times N$. Biết rằng $M\times N$ chia hết cho 6.

- Chứng minh rằng hình chữ nhật này có thể phủ bởi các 3-mino.
- Lập chương trình xây dựng một phủ nào đó.

10. Cho hình vuông $2n\times 2n$ với n không chia hết cho 3. Ta bỏ đi một ô vuông 1×1 bất kì.

- Chứng minh rằng phần còn lại của hình vuông này có thể phủ bởi các 3-mino hình thước thợ.
- Lập chương trình xây dựng một phủ nào đó.

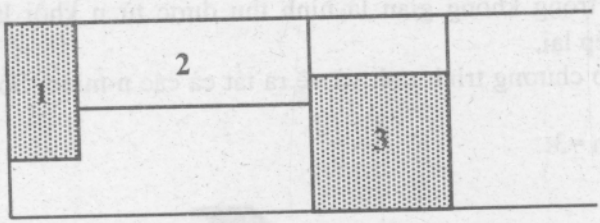
Trong các bài từ 11 đến 18, dữ liệu được đưa vào từ tệp văn bản INPUT.DAT có dạng chuẩn của các dữ liệu cần nhập. Kết quả đưa ra file hoặc màn hình tuỳ theo yêu cầu của từng bài toán.

11. Cho trước k hình vuông con cạnh $A_1, A_2, ..., A_k$ và một hình vuông to cạnh A thoả mãn:

$$A_1^2 + A_2^2 + ... + A_k^2 < A^2$$

Bạn hãy lập chương trình xếp k hình vuông con vào hình vuông to sao cho phần diện tích còn lại có thể đặt được một hình vuông lớn nhất.

12. Giả sử có n mẫu vải hình chữ nhật với kích thước tương ứng là (a_1, b_1), (a_2, b_2), ..., (a_n, b_n). Cần cắt ra từ một băng vải chiều rộng R, chiều dài là vô tận, đúng n mẫu trên sao cho số vải thừa ra có diện tích là nhỏ nhất (chỉ tính số vải thừa ở chỗ bị cắt ra).



13. Hai hình chữ nhật được gọi là không so sánh được nếu không thể xếp hình nọ nằm bên trong hình kia. Cho trước hình chữ nhật m×n. Bạn hãy lập chương trình kiểm tra xem nó có thể phủ được bởi các hình chữ nhật con không so sánh được với nhau hay không. Trong trường hợp có thể hãy chỉ ra một cách phủ như vậy.

14. Dãy các điểm trên mặt phẳng $P_1, P_2, ..., P_n$ được gọi là "gần nhau" nếu tồn tại một hoán vị của dãy trên $Q_1, Q_2, ..., Q_n$ sao cho khoảng cách từ Q_i đến Q_{i+1} không vượt quá 1 với mọi i.
Cho trước dãy điểm $P_1, P_2, ..., P_n$, hãy kiểm tra xem chúng có phải là "gần nhau" hay không?

15. Một đa giác không tự cắt trên mặt phẳng được cho bởi các điểm $A_1, A_2, ..., A_n$. Cho trước điểm M trên mặt phẳng. Hãy viết chương trình kiểm tra xem điểm M nằm trong hay ngoài đa giác trên.

16. Cho một lưới ô vuông kích thước NxM, trên đó có k ô được đánh dấu. Hãy tìm cách tô màu k ô này bằng hai màu Xanh và Đỏ sao cho trên mỗi hàng ngang và hàng dọc của lưới, số lượng các ô Xanh và Đỏ hơn kém nhau không quá đơn vị.

17. Trên mặt phẳng cho trước n hình chữ nhật có các cạnh song song với trục tọa độ C_1, C_2, ..., C_n. Các hình chữ nhật này cho bởi tọa độ hai đỉnh đối diện.

a. Hãy viết chương trình chỉ ra một tập con cực đại các hình chữ nhật lồng nhau của dãy trên.

b. Với mỗi i, hãy chỉ ra một tập con cực đại các hình chữ nhật lồng nhau và chứa C_i.

Bài 13. Các thuật toán sắp xếp

Bài tập

1. Cho mảng N phần tử biểu diễn N quả bóng. Mỗi quả bóng có một trong 3 màu Xanh, Đỏ, Vàng. Hãy dùng các phép đổi vị trí để sắp xếp N quả bóng trên sao cho lần lượt xuất hiện các quả bóng Xanh, sau đó đến Đỏ, cuối cùng là Vàng. Yêu cầu thuật toán nhanh nhất.

2. Cho mảng số thực a_i, a_{i+1}, ..., a_{i+n}. Hãy lập chương trình sắp xếp mảng đó sao cho: Phần đầu của mảng là các phần tử lớn hơn a_i, tiếp đến là chính a_i và các phần tử bằng a_i, sau cùng là các phần tử nhỏ hơn a_i. Yêu cầu thuật toán sử dụng không quá n -1 phép so sánh. Chương trình phải chỉ ra được phần tử a_i đứng ở vị trí thứ mấy của dãy số.

3. Cho mảng số thực a_1, a_2, ..., a_n. Sắp xếp mảng đó bằng thuật toán *nổi bọt*, tức là:

- Bắt đầu từ phần tử đầu tiên, tìm phần tử a_i thoả mãn $a_i > a_{i+1}$ rồi đổi chỗ a_i và a_{i+1}. Sau đó lại tiếp tục từ a_{i+1} ... cho tới phần tử cuối cùng của mảng. Sau bước này, phần tử lớn nhất của mảng đã được xếp vào vị trí cuối cùng.
- Bắt đầu từ phần tử đầu tiên, tìm phần tử a_i thoả mãn $a_i > a_{i+1}$ rồi đổi chỗ a_i và a_{i+1}. Sau đó lại tiếp tục từ a_{i+1}...cho tới phần tử a_{n-1}. Sau bước này, phần tử lớn thứ nhì của mảng đã được xếp vào vị trí thứ n-1.
- Tiếp tục cho tới khi chỉ còn phải kiểm tra một phần tử.

Hãy viết chương trình thể hiện thuật toán đó.

4. Cho mảng số thực a_1, a_2, ..., a_n. Sắp xếp mảng đó bằng thuật toán *chọn*, tức là:

- Trước tiên tìm phần tử nhỏ nhất của mảng, đổi chỗ phần tử này với phần tử đầu tiên. Như vậy sau bước này, phần tử nhỏ nhất của mảng sẽ là phần tử đầu tiên.
- Xét tiếp mảng bắt đầu từ phần tử thứ hai. Tiếp tục thao tác giống như trên. Sau bước này phần tử thứ hai của mảng sẽ được đặt đúng vị trí.
- Tiếp tục các bước trên cho đến khi sắp xếp xong mảng đã cho.

Hãy viết chương trình thể hiện thuật toán trên.

5. Cho mảng số thực $a_1, a_2, ..., a_n$. Sắp xếp mảng đó bằng thuật toán *chèn đơn*, tức là: Lần lượt xét i từ 2 đến n, mỗi phần tử a_i được chèn vào dãy đã sắp xếp $a_1, a_2,..., a_{i-1}$. Vị trí chèn tìm được bằng cách so sánh lần lượt a_i với các phần tử của $a_1, a_2,..., a_{i-1}$. Hãy viết chương trình thể hiện thuật toán đó.

6. Người ta đã chứng minh được rằng, sắp xếp bằng thuật toán chọn có số lần đổi chỗ các phần tử (tức chuyển từ vị trí này qua vị trí khác) ít hơn các thuật toán kia, do đó sử dụng nó sẽ có lợi trong trường hợp việc đổi chỗ các phần tử là rất khó khăn. Hãy vận dụng nhận xét đó trong việc giải bài tập sau:
Cho mảng số thực kích thước m× n. Hãy sắp xếp các hàng (thay đổi vị trí) theo:

a. Thứ tự tăng dần của phần tử đứng đầu hàng.
b. Thứ tự tăng dần của tổng các phần tử của hàng.
c. Thứ tự tăng dần của phần tử nhỏ nhất của hàng.
d. Thứ tự giảm dần của phần tử lớn nhất của hàng.

7. Trong thuật toán chèn đơn, nếu việc tìm vị trí chèn a_i vào dãy $a_1, a_2,..., a_{i-1}$ được thực hiện theo thuật toán tìm kiếm nhị phân thì thuật toán sẽ được gọi là thuật toán *chèn nhị phân*. Hãy viết chương trình thể hiện thuật toán đó.

8. Cho dãy số nguyên $a_1, a_2, ..., a_n$. Hãy thu được dãy các giá trị khác nhau của dãy đã cho được sắp xếp theo thứ tự tăng dần. Yêu cầu dùng thuật toán chèn nhị phân với một cải biên nhỏ: Nếu trong quá trình tìm vị trí chèn a_i mà phát hiện ra rằng trong dãy $a_1, a_2, ..., a_{i-1}$ có giá trị trùng với a_i thì không chèn a_i vào dãy đó nữa.

9. Cho hai dãy số đơn điệu tăng:

$$c_1 \le c_2 \le ... \le c_n \text{ và } d_1 \le d_2 \le ... \le d_m.$$

Hãy viết chương trình để ghép hai dãy trên vào một dãy đơn điệu tăng duy nhất:

$$e_1 \le e_2 \le ... \le e_{n+m}.$$

Yêu cầu dùng không quá n+m phép so sánh.
Thuật toán của bài này được gọi là *thuật toán ghép nối* hay *thuật toán trộn*.

10. Thuật toán trộn mảng dùng để sắp xếp lại mảng số dựa trên ý tưởng thuật toán của bài toán trước như sau:

Trước tiên mảng đã cho được tách thành nhiều đoạn nhỏ, mỗi đoạn nhỏ này đã được sắp thứ tự tăng dần. Gọi k là số các đoạn như vậy. Nếu k = 1 thì dừng chương trình. Thuật toán trộn cho phép làm giảm số k này xuống 1.

- Sử dụng cách làm của bài trên, tiến hành trộn các cặp đoạn 1 và 2, 3 và 4, ...Nếu k lẻ thì đoạn cuối cùng không sử dụng trong bước này.
- Sau bước trên số k sẽ giảm đi khoảng một nửa. Nếu k > 1 thì quay lại thực hiện tiếp bước 2, nếu k = 1 thì kết thúc.

Hãy viết chương trình mô tả thuật toán trộn trên.

11. Thuật toán trộn có thể áp dụng cho việc sắp xếp các phần tử của file. Giả sử cho hai file F và G. Mỗi file này có dạng bản ghi như sau:

```
Type
        NumberFile = Record
              Number: Real;
          End;
```

Từ hai file cho trước như trên, hãy tạo ra một file trộn Q kết quả của hai file trên sau khi đã sắp xếp lại các phần tử của Q theo thứ tự tăng dần. Chương trình không được dùng mảng để mô tả các file này vì số phần tử của các file này có thể rất lớn.

Hãy viết chương trình thể hiện thuật toán trộn file trên.

12. Cho mảng số thực $a_0, a_1, a_2, ..., a_n$. Sắp xếp mảng đó bằng thuật toán *sắp xếp nhanh*, tức là dùng thủ tục đệ qui sau:

<u>Bước 1.</u> Nếu n = 0 thì mảng đã được sắp xếp. Nếu n > 0 thì sắp xếp lại mảng theo yêu cầu giống bài 2.

<u>Bước 2.</u> Chia mảng thành 2 phần tại a_0, với từng phần ta sắp xếp lại như bước 1.

Thuật toán sắp xếp nhanh dùng đệ qui có thể mô phỏng như sau:

```
Procedure quicksort(L,R);
Procedure Sort(L,R);
begin
i:=L; j:=R;
X:=a[(L+R) div 2];
Repeat
  While a[i] < X do i:=i+1;
  While X < a[j] do j:=j-1;
  If i <= j then
```

```
        Đổi chỗ a[i] và a[j]
        i:=i+1;j:=j-1;
        EndIf
    Until i > j;
      If L < j then Sort(L,j);
      If i < R then Sort(i,R);
      end; {Sort}
    BEGIN
      Sort(1,n);
    END {QuickSort }
```

Hãy viết chương trình sinh một dãy số ngẫu nhiên có 10000 phần tử và dùng thuật toán trên để sắp xếp dãy số này.

13. Thuật toán sắp xếp nhanh của bài trên có thể viết lại dưới dạng không đệ qui như sau:

```
{Gán các giá trị ban đầu }
s:=1; stack[1].Left:=1; stack[1].Right:=n;
Repeat
L:=stack[s].Left; R:=stack[s].Right; s:=s-1;
Repeat
 i:=L;j:=R; X:= a[(L+R) div 2];
Repeat
While a[i] < X do i:=i+1;
While a[j] > X do j:=j-1;
If i <= j then
Đổi chỗ a[i], a[j]
i:=i+1; j:=j-1;
EndIf;
Until i > j;
s:=s+1; stack[s].Left:=i;stack[s].Right:=j;
R:=j;
Until L >= R
Until s = 0
```

Hãy viết chương trình sinh một dãy số ngẫu nhiên có 10000 phần tử và dùng thuật toán trên để sắp xếp dãy số này.
So sánh kết quả và tốc độ tính toán với bài trên.

14. Cho trước dãy số thực $a_1, a_2, a_3, ..., a_n$. Hãy viết chương trình tìm ra phần tử đứng thứ k trong dãy trên với k cho trước ($1 \le k \le n$).

15. Cho 3 dãy số thực:

$a_1, a_2, a_3, ..., a_n$

$b_1, b_2, b_3, ..., b_n$

$c_1, c_2, c_3, ..., c_n$

Hãy viết chương trình tìm ra một dãy cực đại các chỉ số j1, j2, ..., jk sao cho:

$a_{j1} \leq a_{j2} \leq ... \leq a_{jk}$

$b_{j1} \leq b_{j2} \leq ... \leq b_{jk}$

$c_{j1} \leq c_{j2} \leq ... \leq c_{jk}$

Bài 14. Các thuật toán tối ưu

Bài tập

1. Cho trước dãy số thực $a_1, a_2, ..., a_n$. Hãy tìm một dãy con đơn điệu tăng có độ dài cực đại của dãy số trên. Yêu cầu thuật toán là tối ưu.

2. Thuật toán 0-1 (hay còn gọi thuật toán ưu tiên 0).
Xây dựng xâu nhị phân không lặp bậc n, chứa mọi xâu con độ dài n và có độ dài nhỏ nhất. Thuật toán phát biểu như sau:

Bước 1. Xây dựng n-1 ký tự 1.

Bước 2. (Ưu tiên 0 nếu có thể) kiểm tra khả năng bổ sung vào bên phải số 0 của dãy mà không phá vỡ tính không lặp. Nếu thành công, thực hiện việc bổ sung này và quay lại bước 2. Nếu không thành công, thực hiện bước tiếp theo.

Bước 3. Xét việc bổ sung vào bên phải số 1 mà không phá vỡ tính không lặp. Nếu thành công, thực hiện việc bổ sung này và quay lại bước 2. Nếu không thành công, dừng lại.

Hãy chứng minh rằng thuật toán trên thực sự sẽ xây dựng được xâu nhị phân không lặp đầy đủ bậc n và có độ dài nhỏ nhất.

3. Cho trước dãy số thực $a_1, a_2, a_3, ..., a_n$. Tổng con theo định nghĩa là tổng có dạng:

$$S_{ij} = a_i + a_{i+1} + a_{i+2} + ... + a_j \quad (\text{với } i < j).$$

Hãy viết chương trình tìm ra một tổng con có giá trị nhỏ nhất. Yêu cầu thuật giải là tối ưu.

4. Xét tập hợp các vé gồm các số có k chữ số, bắt đầu từ 0...0 đến 9...9. Có 100 thùng rác được đánh dấu từ 01 đến 99. Các vé này cần bỏ vào các thùng rác theo nguyên tắc sau:
Mỗi vé chỉ được bỏ vào thùng rác được đánh số bằng con số thu được từ vé sau khi xoá đi k-2 chữ số (không cần liền nhau).

- Hỏi cần ít nhất bao nhiêu thùng để đựng tất cả các vé đó?
- Lập thuật toán mô tả cách bỏ vé vào thùng sao cho số thùng là ít nhất.

5. Có N người, mỗi người biết một thông tin khác nhau. Sau mỗi phút một người có thể gọi điện thoại cho một người khác và thông báo tất cả các thông tin mình biết. Hãy lập cách gọi điện thoại của N người trên để

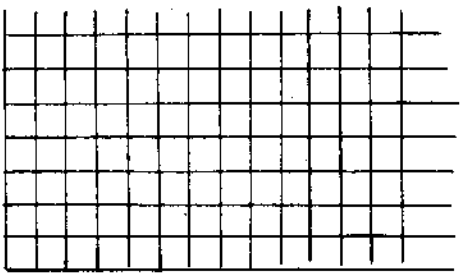
sau ít phút nhất, mỗi người trong N người trên sẽ biết được toàn bộ N thông tin ban đầu.

6. Xâu ký tự S1 được gọi là "nằm trong" xâu ký tự S2 nếu các ký tự trong S1 đều có trong S2 và giữ nguyên thứ tự xuất hiện. Ví dụ xâu "ai" sẽ "nằm trong" xâu "Hanoi".
Cho trước hai xâu ký tự S1, S2. Hãy viết chương trình chỉ ra xâu ký tự dài nhất S3 "nằm trong" cả hai xâu S1 và S2. Yêu cầu thuật toán tối ưu.

7. Có N người ngồi xung quanh một bàn tròn. Cho phép đổi vị trí của hai người bất kỳ ngồi cạnh nhau. Hỏi phải cần bao nhiêu cách biến đổi như vậy và hãy chỉ ra một cách đó sao cho sau khi kết thúc việc đổi chỗ, vị trí của hai người bất kỳ ngồi cạnh nhau trước đây cũng sẽ được "đổi chỗ", tức là nếu như ban đầu A ngồi bên trái B, thì sau đó A sẽ ngồi bên phải của B.

8. Có 2N người ngồi xung quanh một bàn tròn, trong đó mỗi người có không quá N-1 kẻ thù trong số những người còn lại. Được phép đổi vị trí ngồi của hai người bất kỳ trên bàn tròn. Hãy chỉ ra một cách chuyển chỗ tối ưu nhất (dùng ít phép đổi chỗ nhất) sao cho sau đó không có hai người nào là kẻ thù mà lại ngồi cạnh nhau.

9. Cho lưới ô vuông vô hạn về hai phía như hình vẽ:



Các ô của lưới trên được đánh số theo qui tắc sau:
- Ô trái dưới (vị trí 0,0) được đánh số 0.
- Các ô còn lại được đánh số theo nguyên tắc lan tỏa từ vị trí 0,0 và sao cho tại một vị trí số được điền vào là số nguyên không âm nhỏ nhất chưa được điền trên hàng và cột chứa ô hiện thời.

Ví dụ ta có hình dạng của một số ô của lưới trên:

3	2	1	0
2	3	0	1
1	0	3	2
0	1	2	3

Cho trước cặp số tự nhiên M, N, hãy viết chương trình mô tả lưới trên với M hàng và N cột. Kết quả được ghi trong một tệp có tên KQ.TXT.

10. Cho hình vuông H kích thước N×N.

Ta gọi một *phân hoạch* P của H là một cách chia H thành các hình chữ nhật con có tọa độ nguyên và các cạnh song song với các cạnh của hình vuông H.

$$P = \{P_1, P_2, ..., P_k\}$$

Tập con $Q = \{P_{i1}, P_{i2}, ..., P_{is}\}$ được gọi là:

- Phủ, nếu hình chiếu của các phần tử của Q lên một cạnh nào đó của hình vuông sẽ phủ kín cạnh này.
- Phủ đơn, nếu hình chiếu của các phần tử của Q lên một cạnh nào đó của hình vuông tạo thành một phủ không giao nhau của cạnh này.

a. Cho trước hình vuông H và tập k hình chữ nhật. Viết chương trình kiểm tra xem tập các hình chữ nhật này có tạo thành một phân hoạch của hình vuông H hay không?

b. Cho trước một phân hoạch P của hình vuông H và một phần tử A của phân hoạch này. Hãy chỉ ra một phủ nhỏ nhất chứa A.

c. Cho trước một phân hoạch P của hình vuông H và một phần tử A của phân hoạch này. Hãy chỉ ra một phủ đơn nhỏ nhất chứa A.

11. Cho trước hình vuông H kích thước N×N. Hãy viết chương trình sinh ngẫu nhiên các phân hoạch (xem định nghĩa bài trên) của hình vuông này.

12. Bài toán: "Kết bạn bền vững":

Cho trước n bạn trai và n bạn gái. Ta sẽ ký hiệu các bạn trai là m_1, m_2, ..., m_n và các bạn gái sẽ f_1, f_2, ..., f_n. Mỗi bạn (trai hoặc gái) sẽ được quyền đưa ra danh sách các bạn khác giới theo thứ tự ưa thích của mình. Yêu cầu của bài toán là cần ghép thành n cặp bạn (trai - gái) sao cho "tối ưu" nhất. Điều kiện tối ưu hay còn gọi là "Kết bạn bền vững" ở đây hiểu theo nghĩa: Không tồn tại hai bạn trai, gái M, F không cùng cặp với nhau, nhưng M lại "ưa" F hơn bạn cùng cặp và ngược lại F "ưa" M hơn bạn cùng cặp của mình.

Đầu vào là hai ma trận vuông n × n mô tả danh sách các bạn được ưa thích của n bạn trai và n bạn gái. Ví dụ ta có hai ma trận sau (với n = 4):

2 1 4 2	3 4 1 2
1 4 3 3	2 1 3 4
3 2 1 4	4 3 2 1
4 3 2 1	1 2 4 3

Hãy viết chương trình tạo ra một cách chia cặp kết bạn bền vững cho các thông số đầu vào trên.

13. Trò chơi Bốc sỏi.
Có ba đống sỏi và hai người chơi. Mỗi người đến lượt mình được quyền bốc một số sỏi bất kỳ từ một đống. Người bốc được những viên sỏi cuối cùng là người thắng cuộc.
Tìm thuật toán tối ưu cho các người chơi.

14. Trò chơi Hai đống sỏi.
Có hai đống sỏi và hai người chơi. Mỗi người đến lượt mình được quyền:
- Hoặc bốc từ một đống một số sỏi bất kỳ,
- Hoặc lấy đi từ cả hai đống một số sỏi như nhau.

Người đi cuối cùng sẽ là người thắng cuộc.
Tìm thuật toán tối ưu cho các người chơi.

15. Bài toán Hai đống sỏi tổng quát.
Có hai đống sỏi và hai người chơi. Mỗi người đến lượt mình có thể:
- Hoặc bốc từ một đống một số sỏi bất kỳ,
- Hoặc lấy từ một đống k sỏi và đống kia *l* sỏi thỏa mãn $|k-l| \leq$ Const cho trước.

Người đi nước cuối cùng là người thắng cuộc.
Tìm thuật toán tối ưu cho các người chơi.

Bài 15. Các bài toán thi học sinh giỏi

Bài tập

1. *(Đề thi Tin học Quốc tế năm 1994 tại Thụy Điển)*
Hình 1 biểu diễn một tam giác số. Hãy viết chương trình tính tổng tất cả các số trên con đường bắt đầu từ đỉnh và kết thúc đâu đó ở đáy, sao cho tổng này là lớn nhất.

7

3 8

8 1 0

2 7 4 4

4 5 2 6 5

Hình 1

- Mỗi bước có thể đi chéo xuống phía trái hoặc đi chéo xuống phía phải.
- Số lượng hàng trong tam giác lớn hơn 1 nhưng nhỏ hơn hoặc bằng 100.
- Các số trong tam giác đều là số nguyên từ 0 đến 99.

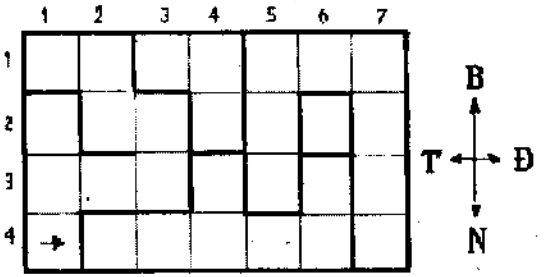
INPUT DATA (dữ liệu vào):
Dữ liệu về số lượng hàng của tam giác được đọc ra đầu tiên từ file INPUT.TXT.
Ở ví dụ file INPUT.TXT là như sau:
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5

OUTPUT DATA (dữ liệu ra):
Tổng lớn nhất như là một số nguyên trong file ra:
OUTPUT.TXT
Với ví dụ, kết quả là: 30

2. *(Đề thi Tin học Quốc tế năm 1994 tại Thụy Điển)*
Hình 2 biểu diễn một bản đồ lâu đài. Hãy viết chương trình tính:

1. Lâu đài có bao nhiêu phòng?
2. Phòng lớn nhất là bao nhiêu?
3. Bức tường nào cần loại bỏ để phòng càng rộng càng tốt?

Lâu đài được chia thành m,n ($m \leq 50$, $n \leq 50$) modul vuông. Mỗi modul vuông có thể có từ 0 đến 4 bức tường.



Hình 2

INPUT DATA
Bản đồ được lưu trữ trong file INPUT.TXT ở dạng các số cho mỗi modul.

- File bắt đầu từ số lượng các modul theo hướng Bắc-Nam và số lượng các modul theo hướng Đông-Tây.
- Trong các phòng tiếp theo mỗi modul được mô tả bởi một số ($0 \leq p \leq 15$). Số đó là tổng của: 1 (= tường phía Tây), 2 (= tường phía Bắc), 4 (= tường phía Đông), 8 (= tường phía Nam).

Các bức tường bên trong được xác định 2 lần, bức tường phía Nam trong modul 1,1 đồng thời là bức tường phía Bắc trong modul 2,1.

- Lâu đài luôn có ít nhất 2 phòng.

INPUT.TXT của ví dụ:
4
7

11 6 11 6 3 10 6
7 9 6 13 5 15 5
1 10 12 7 13 7 5
13 11 10 8 10 12 13
OUTPUT DATA
Trong file ra OUTPUT.TXT viết trên 3 dòng: dòng thứ nhất viết số lượng phòng, dòng tiếp đến là diện tích của phòng lớn nhất (tính theo số modul) và bức tường cần loại bỏ (trước tiên là hàng sau đó là cột của modul có tường đó) và cuối cùng là hướng của bức tường. Trong ví dụ ("4 1 E" là một trong số các khả năng có thể, bạn chỉ cần chỉ ra một).
5
9
4 1 E

3. *(Đề thi Tin học Quốc tế năm 1994 tại Thụy Điển)*

1	1	3	5	1
3	3	2	0	3
3	0	3	2	3
1	4	0	3	3
3	3	3	1	1

Hình 3

Hình 3 biểu diễn một hình vuông. Mỗi dòng, mỗi cột và hai đường chéo có thể đọc thành một số nguyên tố có 5 chữ số. Các dòng đọc từ trái sang phải. Các cột đọc từ trên xuống dưới. Cả hai đường chéo được đọc từ trái sang phải.
Sử dụng dữ liệu trong file INPUT.TXT để viết chương trình xây dựng các hình vuông như vậy.

- Các số nguyên tố phải có cùng tổng các chữ số (11 trong ví dụ trên).
- Chữ số ở đỉnh trái trên của hình vuông được xác định trước (1 trong ví dụ).
- Một số nguyên tố có thể được sử dụng nhiều lần trong cùng một hình vuông.

• Nếu có nhiều nghiệm thì tất cả các nghiệm đó phải được chỉ ra.

• Số nguyên tố có 5 chữ số không được bắt đầu bằng chữ số 0. Ví dụ: 0003 không phải là số nguyên tố có 5 chữ số.

INPUT DATA

Chương trình đọc dữ liệu từ file INPUT.TXT.

Trước tiên là tổng các chữ số của các số nguyên tố, tiếp đến là chữ số đứng ở góc trái trên. File có hai dòng.

Với dữ liệu luôn luôn có nghiệm.

Với ví dụ trên sẽ là:

11

1

OUTPUT DATA

Trong file OUPUT.TXT ghi 5 dòng cho mỗi nghiệm tìm được. Mỗi dòng là một chữ số nguyên tố có 5 chữ số.

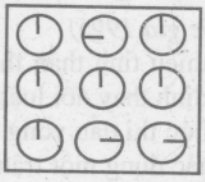
Với ví dụ trên có 3 nghiệm, như vậy file ra OUTPUT.TXT như sau (dòng trống giữa các nghiệm khác nhau là tuỳ ý).

11351
14033
30323
53201
13313
11351
33203
30323
14033
33311
13313
13043
32303
50231
13331

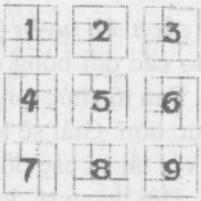
4. *(Đề thi Tin học Quốc tế năm 1994 tại Thuỵ Điển)*

Trong miền 3×3 (h. 4) có 9 chiếc đồng hồ. Mục đích là quay tất cả các kim số của tất cả các chiếc đồng hồ về 12 giờ đúng, sao cho số chuyển dịch càng ít càng tốt. Có 9 cách được phép khác nhau quay các kim số trên các đồng hồ. Mỗi cách như vậy được gọi là một chuyển dịch, chọn

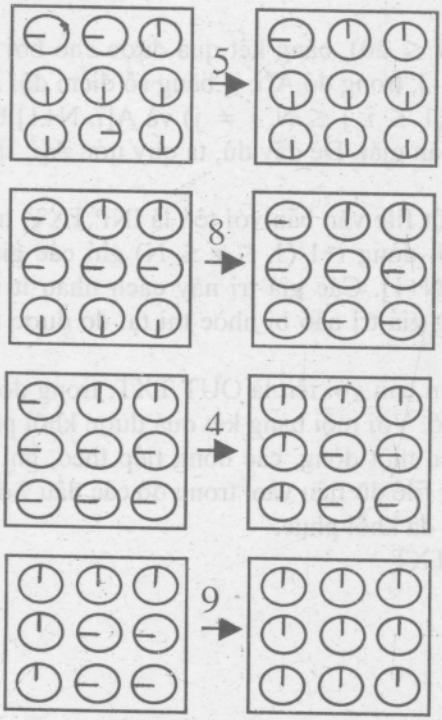
cho mỗi chuyển dịch một số từ 1 đến 9. Số đó sẽ quay kim số 90° theo chiều kim đồng hồ trên các đồng hồ được đánh dấu ở hình 5.



Hình 4



Hình 5



Hình 6

INPUT DATA

Đọc 9 số từ file dữ liệu vào INPUT.TXT. Các số đó chỉ vị trí bắt đầu của kim số. 0 = 12 giờ, 1 = 3 giờ, 2 = 6 giờ và 3 = 9 giờ.

Trong ví dụ ở hình 4, file dữ liệu vào là như sau:

3 3 0
2 2 2
2 1 2

OUTPUT DATA

Ghi vào file dữ liệu ra OUTPUT.TXT một xâu ngắn nhất các dịch chuyển (các số) làm quay tất cả các kim đồng hồ về vị trí 12 giờ. Trong ví dụ cho ở trên, file

OUTPUT.TXT sẽ là: 5 8 4 9

Nếu có nhiều nghiệm chỉ cần đưa ra một nghiệm.

Hình 6 là hình trạng các bước chuyển dịch theo ví dụ.

5. *(Đề thi Tin học chọn đội tuyển Quốc gia 1995)*
Ở một nước nọ do các đội bóng đá thiếu tinh thần thi đấu nên số trận đấu hòa rất nhiều. Ban tổ chức quyết định thay đổi luật để kích thích các đội thi đấu tích cực hơn. Giải sẽ tổ chức thi đấu vòng tròn nghĩa là mỗi đội đều phải đấu với tất cả các đội khác đúng một trận. Nếu hòa, cả hai đội đều bị 0 điểm. Đội thắng sẽ được 3 điểm, còn đội thua cũng sẽ được 1 điểm. Do sơ xuất của ban tổ chức, bảng kết quả thi đấu của tất cả các đội bị nhòe một số chỗ không đọc được. Ban tổ chức quyết định gửi bảng đó đến một trung tâm Tin học nhờ xem xét việc khôi phục lại các thông tin bị mất.
Giả sử số đội bóng bằng N ($N \leq 20$), bảng kết quả được cho bởi một mảng số nguyên A[1..N, 1..N+1], trong đó A[i, j] bằng số điểm đội i đạt được trong trận đấu với đội j ($1 \leq i, j \leq N, i \neq j$) và A[i, N+1] bằng tổng số điểm của đội i trong toàn giải. Để đầy đủ, ta quy ước A[i, i] = 0 ($1 \leq i \leq N$).
Dữ liệu vào được cho trong một file văn bản với tên là INP.TXT, trong đó dòng đầu tiên ghi giá trị N, dòng i+1 ($1 \leq i \leq N$) ghi các giá trị A[i,1], A[i,2], ..., A[i,N], A[i,N+1]. Các giá trị này cách nhau ít nhất một dấu trắng và quy ước rằng giá trị nào bị nhòe thì tại đó được thay bằng dấu ?.
Lời giải cần đưa ra một file văn bản với tên là OUT.TXT, trong đó liệt kê tất cả các khả năng có thể có. Với mỗi bảng kết quả được khôi phục, cần ghi số thứ tự của bảng trên một dòng, các dòng tiếp theo, ghi các dòng của bảng giống như trong file dữ liệu vào, trong đó các dấu ? được thay bằng các giá trị tương ứng đã khôi phục.
Ví dụ với file dữ liệu vào INP.TXT
4
0 1 3 ? 7
? 0 ? ? 6
1 ? 0 1 ?
? ? ? 0 ?
File in kết quả OUT.TXT sẽ là:
1
0 1 3 3 7
3 0 0 3 6

1 0 0 1 2
1 1 3 0 5
2
0 1 3 3 7
3 0 3 0 6
1 1 0 1 3
1 0 3 0 4

6. *(Đề thi Quốc tế IOI lần thứ nhất)*
Cho 2N hộp xếp cạnh nhau. Trong chúng có hai hộp rỗng xếp liền nhau, N-1 hộp chứa chữ A, N-1 hộp chứa chữ B.
Ví dụ cho trường hợp N = 5 (hình 7):

A	B	B	A			A	B	A	B

Hình 7

Luật biến đổi: Hai hộp trống có thể đổi chỗ cho hai hộp liền nhau bất kỳ khác và không làm thay đổi thứ tự của các hộp này.

Mục đích: Thu được vị trí mà tất cả các hộp chứa chữ A nằm về bên trái của các hộp, không để ý đến các hộp rỗng.
Hãy viết chương trình thực hiện:

a. Đọc từ bàn phím trạng thái ban đầu của các hộp với A, B và 0 (thay thế cho các hộp rỗng).
b. Với trạng thái ban đầu trên, tìm ra ít nhất một cách biến đổi để đạt được mục đích hoặc thông báo rằng cách đó không tồn tại. Kết quả đưa ra phải bao gồm trạng thái ban đầu, các trạng thái trung gian sau mỗi bước và trạng thái cuối cùng.
c. Hãy tìm ra cách biến đổi để đạt được mục đích với số bước tối thiểu và đưa ra ít nhất một phương án cho ví dụ đã nêu trên.

7. *(Đề thi Quốc tế IOI lần thứ 2 - vòng 1)*
Cho bảng A gồm 4×4 ô. Mỗi ô được điền các số từ 1 đến 14, mỗi số điền một lần. Hai ô còn lại để trống.
Ví dụ bảng A cho ở hình 8.

1		5	6
2	4	9	11
	8	7	14
3	10	13	12

Hình 8

1	2	3	4
5	6	7	8
9	10	11	12
13	14		

Hình 9

Mục đích: Biến đổi bảng A bất kỳ thành bảng B kết thúc (h. 9).
Quy tắc biến đổi: Một số ở một ô bất kỳ có thể được dịch chuyển theo chiều ngang hoặc thẳng đứng sang ô bên cạnh nếu ô này trống.
Yêu cầu:
Lập chương trình thực hiện các bước sau :

a. Nhập số liệu cho bảng A (nhập theo từng dòng).
b. Biến đổi bảng A để trở thành bảng kết thúc B (h.9).

Mỗi bước biến đổi in ra màn hình:

- Số thứ tự của lần chuyển.
- Bên trái màn hình là bảng trước khi chuyển.
- Bên phải màn hình là bảng sau khi chuyển.
- Cuối cùng in ra tổng số lần chuyển để có bảng B.

c. Tìm cách chuyển sau ít bước chuyển nhất.

8. *(Đề thi Tin học Quốc tế năm 1994 tại Thụy Điển)*

Một người đến một bến đỗ xe buýt vào lúc 12 giờ 00. Người đó ở đó suốt từ 12 giờ 00 đến 12 giờ 59. Bến đỗ là của một số tuyến xe buýt. Các thời gian đến bến của các xe được cho.

- Các xe trên cùng một tuyến đường đến bến đỗ theo các khoảng đều đặn từ 12 giờ 00 đến 12 giờ 59 (suốt nguyên cả giờ).
- Các thời điểm được cho nguyên phút từ 0 đến 59.
- Xe mỗi tuyến đường đỗ ít nhất 2 lần.
- Số lượng các tuyến đường trong các ví dụ kiểm tra ≤ 17.
- Các xe từ các tuyến khác nhau có thể đến bến cùng thời gian.
- Một số tuyến xe khác nhau có thể có thời gian đến bến lần đầu như nhau và có thể có cùng khoảng thời gian đến như nhau.

Nếu có hai tuyến xe có cùng thời gian khởi hành và cùng khoảng thời gian đến thì phải được phân biệt để cả hai tuyến xe đó đều được thể hiện.
Tìm số lượng ít nhất các tuyến xe buýt phải đỗ lại ở bến đỗ để thoả mãn

dữ liệu vào.
Với mỗi tuyến xe buýt đưa ra thời gian bắt đầu và khoảng thời gian đến.

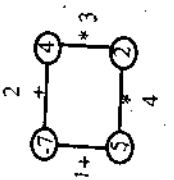
INPUT DATA
File dữ liệu vào INPUT.TXT chứa n ($n \leq 300$) là số lượng xe đến bến ghi được, tiếp đó là thời gian đến bến theo thứ tự tăng.
Ví dụ:
17
0 3 5 13 13 15 21 26 27 29 37 39 39 45 51 52 53

OUTPUT DATA
Ghi một bảng vào file ra OUTPUT.TXT mỗi dòng cho mỗi tuyến xe.
Mỗi dòng cho: thời gian đến của xe đầu tiên, khoảng thời gian đến tính theo phút. Chỉ yêu cầu tìm ra một nghiệm.
Ví dụ trên cho:
0 13
3 12
5 8

Thứ tự các tuyến đường là không quan trọng.

9. Trò chơi xoá cạnh đa giác

(*Polygon - Đề thi Olympic IOI-1998*)
Polygon là một trò chơi cho một người mà bắt đầu bởi một đa giác N đỉnh, giống như hình 1, với N = 4. Mỗi đỉnh được đánh nhãn với một số nguyên và mỗi cạnh được đánh nhãn bằng hoặc là dấu cộng hoặc là dấu *. Các cạnh được đánh số từ 1 đến N.



Hình 10

Trong bước đi đầu tiên, một trong các cạnh được tách ra.
Sau đó lần lượt đi theo các bước sau:

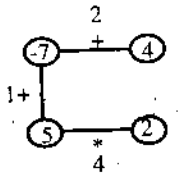
- Chọn một cạnh E và 2 đỉnh V_1 và V_2 là hai đầu mút của cạnh E.

- Thay thế chúng bởi một đỉnh mới là kết quả của việc biểu diễn phép toán biểu thị ở cạnh E của các số được đánh nhãn ở V_1 và V_2.

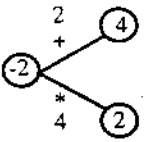
Trò chơi kết thúc khi không còn một cạnh nào, và điểm được tính là nhãn của một đỉnh còn lại.

Ví dụ:

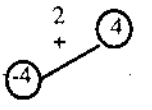
Xét một đa giác như hình 10. Người chơi bắt đầu bằng cách tách cạnh 3 (h. 11). Sau đó, chọn cạnh 1 (h.12) rồi đến cạnh 4 (h.13)



Hình 11



Hình 12



Hình 13

Hình 14

Cuối cùng chọn cạnh 2 (h.14). Điểm thu được trong trò chơi theo cách này là 0.

Nhiệm vụ

Cho đa giác, viết chương trình tính điểm cao nhất có thể và danh sách tất cả các cạnh tách đầu tiên mà nếu dựa vào việc tách cạnh đầu tiên đó thì có thể dẫn đến điểm cao nhất trên.

Dữ liệu vào: trong file POLYGON.IN miêu tả đa giác với N đỉnh, gồm 2 dòng: dòng đầu tiên là N - số đỉnh của đa giác; dòng thứ 2 nhãn của các cạnh 1, ..., N xen lẫn với nhãn của các đỉnh (đầu tiên là đỉnh giữa hai cạnh 1 và 2, sau đó là đỉnh giữa hai cạnh 2 và 3, ..., cuối cùng là đỉnh giữa hai cạnh N và 1), tất cả cách nhau bởi một dấu cách. Nhãn của một cạnh hoặc là *t* (tương ứng với +) hoặc là *x* (tương ứng với *).

Ví dụ đầu vào:

4

t -7 t 4 x 2 x 5

Đây là dữ liệu vào cho đa giác tương ứng với hình 10. Dòng thứ nhất là số đỉnh của đa giác; dòng thứ 2 bắt đầu với nhãn của cạnh 1.

Dữ liệu ra: trong file POLYGON.OUT, dòng đầu phải ghi được số điểm cao nhất nhận được từ dữ liệu vào; dòng thứ 2 là danh sách của tất cả các cạnh được tách đầu tiên mà nếu dựa vào việc tách cạnh đầu tiên đó thì có thể dẫn đến số điểm cao nhất đó.

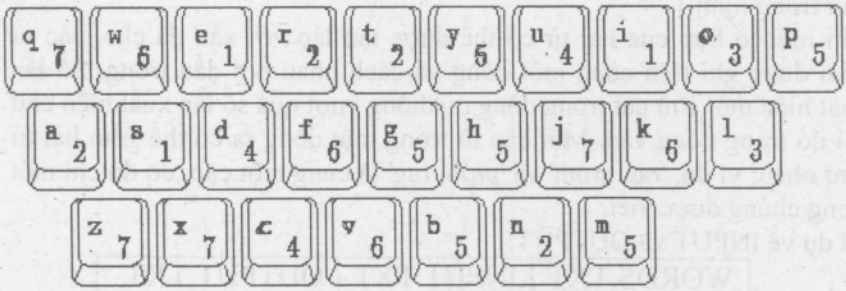
Các cạnh phải được viết theo thứ tự tăng dần, cách nhau bởi dấu cách.

Ví dụ đầu ra
33
1 2
Kết quả này tương ứng với đa giác cho bởi hình 10.
Điều kiện của bài toán:

$3 \leq N \leq 50$ (số đỉnh của đa giác)

Nhãn để đánh dấu các đỉnh là các số nguyên có giới hạn trong đoạn [-32768,32767].

10. Trò chơi chữ
(*LGame - Đề thi Olympic Quốc tế IOI-1995*)



Hình 15. Các chữ cái và giá trị của chúng

Các trò chơi chữ rất phổ biến ở nhà và trên Tivi. Trong một cách chơi, mỗi chữ có một giá trị, và bạn tập hợp các chữ cái để lập thành một hay nhiều từ cho giá trị (điểm) cao nhất có thể được. Mặc dù đã tìm được một cách tạo lập từ nào đó, bạn vẫn phải thử mọi từ mà bạn biết, thỉnh thoảng phải soát lỗi chính tả và sau đó tính giá trị (điểm). Rõ ràng điều đó có thể làm một cách chính xác hơn nhờ máy tính.

Cho các giá trị trong hình 15, một danh sách các từ tiếng Anh và một dãy các chữ cái để tạo lập các từ, hãy tìm các từ hoặc các cặp từ với giá trị cao nhất có thể lập được.

Dữ liệu vào

File Dữ liệu vào INPUT.TXT gồm một dòng chứa một xâu gồm các chữ cái thường (từ 'a' đến 'z'): dãy các chữ cái để tạo lập các từ. Xâu này gồm ít nhất 3 và nhiều nhất 7 chữ cái theo thứ tự bất kỳ.
File 'từ điển' WORDS.TXT gồm nhiều nhất 40000 dòng. Cuối file là một dòng với một dấu chấm duy nhất. Mỗi một trong các dòng khác chứa một xâu ít nhất 3 và nhiều nhất 7 chữ cái thường.
File WORDS.TXT không chứa các từ trùng lặp. Các từ trong file được sắp xếp theo vần a, b, c...

Dữ liệu ra

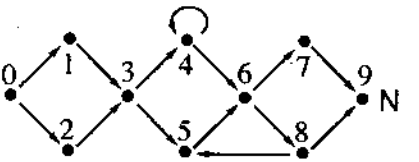
Trên dòng thứ nhất của file OUTPUT.TXT, chương trình của bạn phải ghi giá trị cao nhất (Câu A), và trên mỗi dòng tiếp theo, mọi từ hoặc cặp từ thuộc file WORDS.TXT có giá trị đó (Câu B). Dùng các giá trị của chữ trong hình 1.
Khi một tổ hợp của hai từ có thể được tạo lập với xâu đã cho, các từ phải được ghi trên cùng một dòng và cách nhau bởi dấu trống. Số lần xuất hiện mỗi chữ cái trong dòng ra không vượt quá số lần xuất hiện chữ cái đó trong dòng vào. Một cặp từ trong một dòng ra có thể gồm hai từ như nhau; ví dụ, 'rag prom' và 'prom rag' là cùng một cặp, do đó chỉ một trong chúng được viết.
Ví dụ về INPUT và OUTPUT:

WORDS.TXT	INPUT.TXT	OUTPUT.TXT
profile program prom rag ram rom	prmgroa	24 program rag prom

11. Cuộc dạo phố

(*Race - Đề thi Olympic Quốc tế IOI-1995*)

Hình 16 cho ví dụ về một bản đồ dùng cho một cuộc dạo phố. Bạn thấy số đỉnh đánh số từ 0 đến N (ở đây N = 9) và một số mũi tên nối chúng.



Hình 16. Bản đồ phố với 10 điểm

Đỉnh 0 là đỉnh xuất phát của hành trình, đỉnh N là đỉnh kết thúc. Các mũi tên biểu thị các phố một chiều. Người tham gia cuộc dạo đi từ đỉnh này đến đỉnh kia theo các phố chỉ theo hướng các mũi tên. Tại mỗi đỉnh, người đó có thể chọn một mũi tên bất kỳ để đi theo.

Một bản đồ lập tốt có các tính chất sau:

1. Từ đỉnh xuất phát có thể đi đến mọi đỉnh trên bản đồ.
2. Từ một đỉnh bất kỳ có thể đi đến đỉnh kết thúc.
3. Không có mũi tên ra từ đỉnh kết thúc.

Người tham gia không nhất thiết phải đi qua mọi đỉnh để đi đến đỉnh kết thúc. Tuy nhiên có một số đỉnh không tránh được. Trên ví dụ, các đỉnh này là 0, 3, 6, 9. Cho một bản đồ lập tốt, chương trình của bạn cần xác định tập hợp các đỉnh không tránh được mà mọi người tham gia đều phải đi qua ngoại trừ các đỉnh xuất phát và kết thúc. (Câu A).

Giả sử rằng cuộc dạo chơi diễn ra trong hai ngày liên tiếp. Với mục đích đó, bản đồ chia thành hai bản đồ, mỗi bản đồ cho một ngày. Trong ngày thứ nhất, đỉnh xuất phát là 0 và đỉnh kết thúc là một đỉnh chia cắt nào đó. Ngày thứ hai đỉnh xuất phát là đỉnh chia cắt và đỉnh kết thúc là đỉnh N. Cho một bản đồ lập tốt, chương trình của bạn cần xác định tập hợp các đỉnh chia cắt (Câu B).

Đỉnh S là đỉnh chia cắt đối với bản đồ lập tốt C nếu S khác với đỉnh xuất phát và đỉnh kết thúc của C và bản đồ được chia thành hai bản đồ lập tốt không có mũi tên chung và chỉ có S là đỉnh chung duy nhất.

Dữ liệu vào

File INPUT.TXT mô tả một bản đồ lập tốt với nhiều nhất 50 đỉnh và nhiều nhất 100 mũi tên. Có N+1 dòng trong file.

Trong N dòng đầu, dòng thứ i, $1 \le i \le N$ ghi các đỉnh cuối của các mũi tên xuất phát từ đỉnh i-1. Mỗi dòng này kết thúc bằng số -2. Dòng cuối cùng ghi số -1.

Dữ liệu ra

Chương trình của bạn phải viết ra hai dòng trong file OUTPUT.TXT. Dòng thứ nhất ghi số lượng đỉnh không tránh được trong bản đồ cho bởi file INPUT, tiếp theo là tên các đỉnh không tránh được theo một thứ tự tuỳ ý (Câu A). Dòng thứ hai ghi số lượng các đỉnh chia cắt trong bản đồ cho bởi file INPUT, tiếp theo là tên các đỉnh này theo một thứ tự tuỳ ý (Câu B). (Nếu không có đỉnh chia cắt, chỉ ghi số 0).

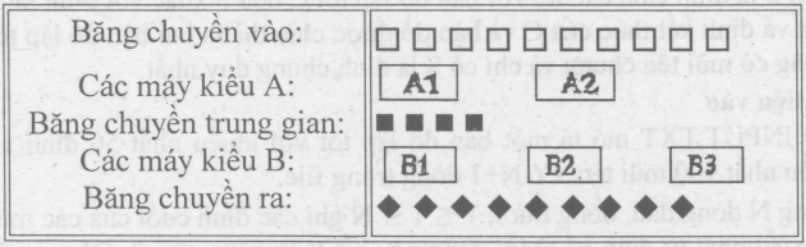
Ví dụ về INPUT và OUTPUT:

INPUT.TXT	OUTPUT.TXT
1 2 -2 3 -2 3 -2 5 4 -2 6 4 -2 6 -2 7 8 -2 9 -2 5 9 -2 -1	2 3 6 1 3

12. Dây chuyền sản xuất

(*Job Processing - Đề thi Olympic Quốc tế IOI-1996*)

Một nhà máy chạy một dây chuyền sản xuất. Có hai nguyên công cần phải thực hiện đối với mỗi sản phẩm theo trình tự sau: đầu tiên là nguyên công "A", sau đó tới nguyên công B. Có một số máy để thực hiện từng nguyên công. Hình 17 chỉ ra cách tổ chức dây chuyền sản xuất hoạt động như sau:



Hình 17

Máy kiểu "A" lấy sản phẩm từ băng chuyền vào, thực hiện nguyên công "A" và đặt sản phẩm vào băng chuyền trung gian. Máy kiểu "B" lấy sản phẩm từ băng chuyền trung gian, thực hiện nguyên công "B" và đặt sản phẩm vào băng chuyền ra. Mọi máy đều có thể làm việc song song và độc lập với nhau. Kích thước của các băng chuyền là không giới hạn. Các máy có khả năng làm việc khác nhau, mỗi máy làm việc có thời gian xử lý cho trước. Thời gian xử lý là số đơn vị thời gian cần thiết để thực hiện nguyên công bao gồm cả thời gian lấy sản phẩm từ băng chuyền trước khi xử lý và thời gian đặt sản phẩm vào băng chuyền sau khi bị xử lý.

Đưa ra thời điểm sớm nhất mà nguyên công "A" được hoàn thành đối với tất cả N sản phẩm với điều kiện là các sản phẩm này đã sẵn sàng trên băng chuyền vào tại thời điểm 0 (Câu A). Đưa ra thời điểm sớm nhất mà cả hai nguyên công "A" và "B" được hoàn thành đối với tất cả N sản phẩm khi các sản phẩm này đã sẵn sàng trên băng chuyền vào tại thời điểm 0 (Câu B).

Dữ liệu vào

File INPUT.TXT gồm các số nguyên dương ghi trong 5 dòng. Dòng thứ nhất chứa N là số sản phẩm ($1 \le N \le 1000$). Trên dòng thứ hai ghi M1 là số lượng các máy kiểu "A" ($1 \le M1 \le 30$). Trên dòng thứ ba ghi M1 số nguyên là các thời gian xử lý của từng máy kiểu "A". Các dòng thứ tư và thứ năm tương ứng ghi M2 là số lượng các máy kiểu "B" ($1 \le M2 \le 30$) và các thời gian xử lý của từng máy kiểu "B". Thời gian xử lý là một số nguyên nằm trong khoảng từ 1 đến 20.

Dữ liệu ra

Chương trình của bạn cần ghi hai dòng ra file OUTPUT.TXT. Dòng đầu tiên chứa một số nguyên dương là lời giải của câu A. Dòng thứ hai chứa lời giải của câu B.

Ví dụ về dữ liệu vào và ra

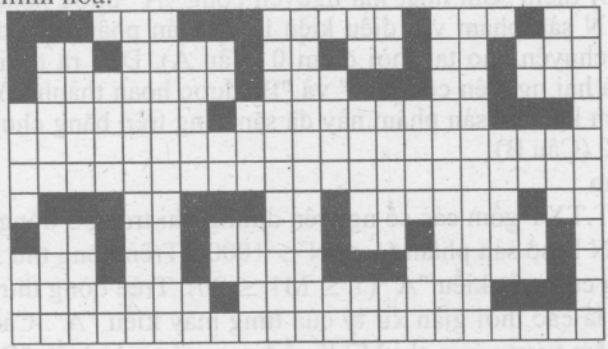
Hình sau cho một file Input có thể có và file Output tương ứng với nó.

INPUT.TXT	OUTPUT.TXT
5	3
2	5
1 1	
3	
3 1 4	

13. Những chòm sao

(*Starry Night - Đề thi Olympic Quốc tế IOI-1998*)

Hãy nhìn lên trời đêm, những ngôi sao sáng ngời xuất hiện thành từng chòm với những hình thù rất khác nhau. Một chòm sao là một tập hợp khác rỗng của những ngôi sao "láng giềng", gần kề nhau theo các chiều ngang, dọc hoặc chéo. Một chòm sao không thể là một phần của một chòm sao lớn hơn. Những chòm sao có thể là giống nhau. Hai chòm sao được gọi là giống nhau nếu chúng có cùng hình dạng và cùng số lượng các ngôi sao (không kể phương hướng của chúng). Thông thường, số phương hướng có thể cho các chòm sao giống nhau này là 8, hình 18 là một ví dụ minh hoạ.



Hình 18. Tám chòm sao giống nhau.

Bầu trời đêm có thể được hình dung như là một bản đồ sao về bầu trời đó, là một ma trận hai chiều gồm các phần tử là các số 0 và 1. Các ô là 1 nếu có sao, ngược lại là 0 nếu không có sao.

Nhiệm vụ

Cho một bản đồ sao, đánh dấu tất cả các chòm sao bằng các chữ cái thường. Các chòm sao giống nhau phải được đánh dấu cùng một chữ cái thường, còn các chòm khác nhau phải được đánh dấu bởi các chữ cái thường khác nhau. Đánh dấu chòm sao với chữ cái thường bằng cách: thay thế mọi số 1 trong chòm sao bằng chữ cái thường, còn số 0 vẫn giữ nguyên.

Dữ liệu vào: trong file STARRY.INP gồm:

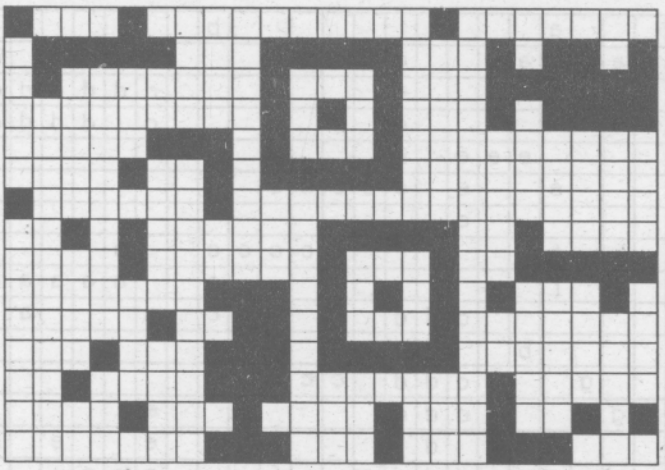
Hai dòng đầu tiên tương ứng với chiều rộng W và chiều cao H của kích thước bản đồ sao (ma trận hai chiều). H dòng tiếp theo, mỗi dòng là W kí tự không cách nhau (là các số 0 hoặc 1: là 1 nếu tương ứng với ô đó là có sao, còn là 0 nếu không có sao).

Ví dụ Input

23
15
100010000000000100000000
011111000111110001011 01
010000000100001000111111
000000000101010001011 11
000001110100010000000000
000010010111110000000000
100000010000000000000000
001010000001111100100 00
000010000001000100111 11
000000011101010101000 10
000001001101000100000 00
000100011101111100000 00
001000011100000000100 000
000010001000010001001 01
000000011100010001110 00

Trong ví dụ này, bản đồ sao có kích thước: chiều rộng là 23 và chiều cao là 15.

Dữ liệu vào phải phù hợp với bức tranh của bản đồ sao (h.19):



Hình 19. Bức tranh về bầu trời sao

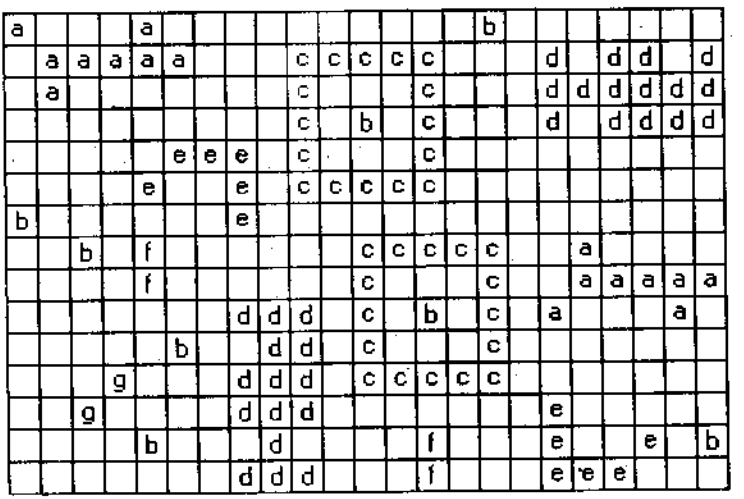
Dữ liệu ra: trong file STARRY.OUT chứa một bản đồ tương tự (ma trận hai chiều với kích thước như thế), nhưng các chòm sao bây giờ được đánh dấu bởi nhiệm vụ trên (tức đã thay số 1 bởi các chữ cái thường, các chòm giống nhau thì được thay bởi kí tự giống nhau).

Ví dụ Output

```
a000a0000000000b0000000
0aaaaa000ccccc000d0dd0d
0a0000000c000c000dddddd
000000000c0b0c000d0dddd
00000eee0c000c000000000
0000e00e0ccccc000000000
b000000e000000000000000
00b0f000000ccccc00a0000
0000f000000c000c00aaaaa
0000000ddd0c0b0c0a000a0
00000b00dd0c000c0000000
000g000ddd0ccccc0000000
00g0000ddd0000000e00000
0000b000d0000f000e00e0b
0000000ddd000f000eee000
```

Kết quả này là tương ứng với dữ liệu vào ở trên.

Chú ý, file kết quả phải phù hợp với bức tranh sau:



Hình 20. Hình ảnh các chòm sao được đánh dấu

Điều kiện:

$0 \le W \le 100$ (chiều rộng).

$0 \le H \le 100$ (chiều cao).

$0 \le$ số chòm sao ≤ 500.

$0 \le$ số các chòm sao không giống nhau ≤ 26 (a..z).

$1 \le$ số ngôi sao của một chòm sao ≤ 160.

14. Mạng các trường học

(*Network of Schools - Đề thi Olympic Quốc tế IOI-1996*)

Một số trường học được nối với nhau bởi một mạng máy tính. Có một sự thoả thuận giữa các trường học này: mỗi trường học có một danh sách các trường học (gọi là danh sách các trường "nhận") và mỗi trường khi nhận được một phần mềm từ một trường khác trong mạng hoặc từ bên ngoài, cần phải chuyển phần mềm nhận được cho các trường trong danh sách các trường nhận của nó. Cần chú ý rằng nếu B thuộc danh sách các trường nhận của trường học A thì A không nhất thiết phải xuất hiện trong danh sách các trường nhận của trường học B.

Người ta muốn gửi một phần mềm đến tất cả các trường học trong mạng. Bạn cần viết chương trình tính số ít nhất các trường học cần gửi bản sao của phần mềm này để cho phần mềm đó có thể chuyển tới tất cả các trường học trong mạng theo thoả thuận trên (Câu A). Ta muốn chắc chắn rằng khi bản sao phần mềm được gửi đến một trường học bất kỳ, phần mềm này sẽ được chuyển tới tất cả các trường học trong mạng. Để đạt mục đích này, ta có thể mở rộng danh sách các trường nhận bằng cách thêm vào các trường mới. Tính số ít nhất các mở rộng cần thực hiện sao cho khi ta gửi một phần mềm mới đến một trường bất kỳ trong mạng, phần mềm này sẽ được chuyển đến tất cả các trường khác (Câu B). Ta hiểu một mở rộng là việc thêm một trường mới vào trong danh sách các trường nhận của một trường học nào đó.

Dữ liệu vào

Dòng đầu tiên của file INPUT.TXT chứa số nguyên N: số trường học trong mạng ($2 \le N \le 100$).Các trường được đánh số bởi N số nguyên dương đầu tiên. Mỗi một trong N dòng tiếp theo mô tả một danh sách các trường nhận. Dòng thứ i+1 chứa số hiệu các trường nhận của trường i.

Mỗi danh sách kết thúc bởi số 0. Dòng tương ứng với danh sách rỗng chỉ chứa một số 0.

Dữ liệu ra

Chương trình của bạn cần ghi hai dòng ra file OUTPUT.TXT. Dòng thứ nhất ghi một số nguyên dương là lời giải của câu A. Dòng thứ hai ghi lời giải của câu B.

Ví dụ về dữ liệu vào và ra:

INPUT.TXT	OUTPUT.TXT
5	1
2 4 3 0	2
4 5 0	
0	
0	
1 0	

15. Sắp xếp dãy với 3 giá trị

(*Sort3 - Đề thi Olympic Quốc tế IOI - 1996*)

Sắp xếp là một trong những công việc tính toán hay phải làm nhất. Xét bài toán sắp xếp cụ thể sau đây: Cần sắp xếp các bản ghi có nhiều nhất ba giá trị của khoá, trong đó các bản ghi cần sắp xếp có nhiều nhất ba giá trị khoá khác nhau. Chẳng hạn, bài toán như vậy sẽ phải giải quyết khi ta muốn sắp xếp những người được huy chương trong một cuộc thi theo giá trị của huy chương mà họ nhận được, có nghĩa là những người đạt huy chương vàng xếp trước, tiếp theo là huy chương bạc và cuối cùng là những người đạt huy chương đồng.

Trong bài này ba giá trị có thể có của khoá là các số nguyên 1, 2 và 3. Cần tiến hành sắp xếp theo thứ tự không giảm của giá trị khoá. Việc sắp xếp cần được thực hiện bằng một dãy các thao tác đổi chỗ. Một thao tác đổi chỗ được xác định bởi hai số p và q chỉ rõ cần đổi chỗ hai phần tử ở vị trí p và q cho nhau.

Bạn được cho một dãy các giá trị khoá. Hãy viết chương trình tính số ít nhất các thao tác đổi chỗ cần thực hiện để sắp xếp dãy giá trị khoá đã cho thành một dãy không giảm (Câu A). Đồng thời, hãy xây dựng dãy các thao tác đổi chỗ tương ứng với cách sắp xếp tìm được. (Câu B).

Dữ liệu vào

Dòng đầu tiên của file INPUT.TXT chứa N là số lượng bản ghi ($1 \le N \le 1000$). Mỗi một trong số N dòng tiếp theo chứa một giá trị khoá.

Dữ liệu ra

Ghi ra dòng đầu tiên của file OUTPUT.TXT số lượng nhỏ nhất L các thao tác đổi chỗ cần thực hiện để sắp xếp dãy đã cho thành dãy không giảm (Câu A). N dòng tiếp theo ghi dãy các thao tác đổi chỗ tương ứng theo trình tự thực hiện. Mỗi dòng chứa một thao tác đổi chỗ được mô tả bởi hai số p và q là các vị trí của hai phần tử được đổi chỗ cho nhau (Câu B). Các vị trí được đánh số từ 1 đến N.

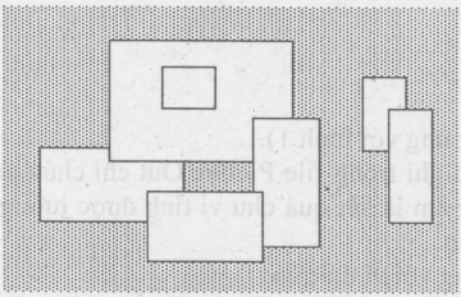
Ví dụ dữ liệu vào và ra:

INPUT.TXT	OUTPUT.TXT
9	4
2	1 3
2	4 7
1	9 2
3	5 9
3	
3	
2	
3	
1	

16. Những bức tranh

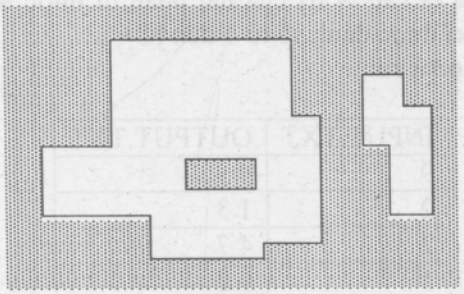
(*Đề thi Olympic Quốc tế IOI-1998*)

Một số tờ áp phích, ảnh để quảng cáo có dạng hình chữ nhật và một số bức tranh khác cũng có hình dạng tương tự được dán lên tường theo chiều ngang hoặc dọc. Mỗi hình chữ nhật có thể là không được hoàn chỉnh hoặc chúng hoàn toàn bị che phủ bởi những hình chữ nhật khác.



Hình 21

Chiều dài của biên hợp bởi tất cả các hình chữ nhật được gọi là chu vi. Bạn hãy lập một chương trình để tính chu vi đó.
Một ví dụ với 7 hình chữ nhật được chỉ ra trên hình 21.
Biên tương ứng là toàn bộ các tập đoạn thẳng được vẽ ở hình 22.



Hình 22

Các đỉnh của tất cả các hình chữ nhật đều có toạ độ nguyên.
Dữ liệu vào: trong file Picture.Inp, dòng đầu tiên là một số nguyên dương N ($0 \leq N \leq 500$) - là số lượng các hình chữ nhật được dán lên tường. N dòng tiếp theo, mỗi dòng lần lượt chứa 4 số nguyên là các toạ độ của đỉnh trái trên và đỉnh phải dưới của các hình chữ nhật đó.Giá trị của các toạ độ này được cho theo thứ tự như cặp toạ độ (x,y) ($-1000 \leq x,y \leq 1000$).

Ví dụ:
7
-15 0 5 10
-5 8 20 25
15 -4 24 14
0 -6 16 4
2 15 10 22
30 10 36 20
34 0 40 16
(Ví dụ này tương ứng với hình 1).
Dữ liệu ra: được ghi trong file Picture.Out chỉ chứa một dòng với một số nguyên không âm là kết quả chu vi tính được tương ứng với dữ liệu đã nhập vào ở trên.
Kết quả tương ứng với ví dụ trên:
228

PHẦN II. CHƯƠNG TRÌNH

Bài 1. Thủ tục và hàm số

9. {$R+,B-}

```
Program P30109; {Gian uoc cac phan so}
Uses Crt;
Type   ps=record
            ts,ms: integer;
          end;
Const   fi='P30109.inp';
        fo='P30109.out';
Var   f,g: file of ps;
      so: ps;
Function USCLN(a,b: integer): integer;
Var r: integer;
Begin
While b<>0 do
  begin
       r:=a mod b;
       a:=b; b:= r;
  end;
USCLN:=a;
End;
Procedure Gianuoc(c: integer; Var a,b: integer);
Begin
 a:=a div c; b:=b div c;
End;
BEGIN
 Assign(f,fi); Reset(f);
 Assign(g,fo); Rewrite(g);
 While not Eof(f) do
   begin
        Read(f,so);
        Write(so.ts,'/',so.ms,' = ');
        Gianuoc(USCLN(so.ts,so.ms),so.ts,so.ms);
        Write(g,so);
        Writeln(so.ts,'/',so.ms);
   end;
 Close(g); Close(f);
```

```
 Readln;
END.
```

11. Program P30111; {Tinh chu vi va duong kinh cua mot da giac}

```
Uses Crt;
Const   fi = 'POLYGON.DAT';
Var   f: text;
      n,i,j: integer;
      point: array[1..1000,1..2] of real;
      cv,bk: real;

Function distance(i,j: integer): real;
Begin
 distance:=sqrt(sqr(point[i,1]-point[j,1])+sqr(point[i,2]-point[j,2]));
End;

BEGIN
 Clrscr;
 Assign(f,fi); Reset(f);
 Readln(f,n);
 For i:=1 to n do readln(f,point[i,1],point[i,2]);
 Close(f);
 cv:=distance(n,1);
 For i:=1 to n-1 do cv:=cv+distance(i,i+1);
 Writeln(' Chu vi = ',cv:10:2);
 bk:=0;
 For i:=1 to n-1 do
  For j:=i+1 to n do
    If distance(i,j)>bk then bk:=distance(i,j);
 Writeln(' Ban kinh = ',bk:10:2);
 Readln;
END.
```

13. Program P30113; {Tinh dinh thuc cua ma tran}

```
Uses Crt;
Type  mt = array[1..10,1..10] of real;
Var a: mt;
     n: byte;
Function Sign(h: byte): integer;
Begin
 If Odd(h) then sign:=-1 Else sign:=1;
End;
```

```
Procedure Nhap;
Var i,j: byte; f: text;
Begin
 Assign(f,'IN.DAT'); Reset(f);
 Readln(f,n);
 For i:=1 to n do
   begin
         For j:=1 to n do Read(f,a[i,j+1]);
         Readln(f);
    end;
 Close(f);
 For i:=1 to n do
   begin
         For j:=1 to n do Write(a[i,j+1]:3:0);
         Writeln;
   end;
End;
Function Det(b: mt; m,s: byte): real;
Var p,q: byte; d: real;
Begin
 For p:=1 to m do
   For q:=1 to m do
     If p<s then b[p,q]:=b[p,q+1]
       Else b[p,q]:=b[p+1,q+1];
 d:=0;
 If m=1 then d:=b[1,1]
   Else For p:=1 to m do d:=d-sign(p)*b[p,1]*det(b,m-1,p);
 Det:=d;
End;

BEGIN
 Clrscr;
 Nhap;
 Write(' Det A = ', det(a,n,n+1):10:0);
 Readln;
END.
```

15. Program P30115; {Tim ma tran nghich dao}
Uses Crt;

```
Type mt = array [1..10,1..10] of real;
Var a,c: mt;
     i,j,n: byte;
    d: real;
Function Sign(h: byte): integer;
Begin
 If Odd(h) then Sign:=-1 Else Sign:=1;
End;
Procedure Nhap;
Var f: text;
Begin
 Assign(f,'IN.DAT'); Reset(f);
 Readln(f,n);
 For i:=1 to n do
  begin
        For j:=1 to n do Read(f,a[i,j]);
        Readln(f);
   end;
 Close(f);
 Writeln('Ma tran A:');
 For i:=1 to n do
   begin
        For j:=1 to n do Write(a[i,j]:6:2);
        Writeln;
   end;
End;
Procedure Xuat;
Var f: text;
Begin
 Assign(f,'OUTPUT.DAT'); Rewrite(f);
 Writeln(f,n);
 For i:=1 to n do
  begin
       For j:=1 to n do Write(f,c[i,j]:6:2);
       Writeln(f);
  end;
 Close(f);
 Writeln('Ma tran nghich dao cua A: ');
 For i:=1 to n do
```

```
  begin
       For j:=1 to n do Write(c[i,j]:6:2);
       Writeln;
   end;
 Readln;
End;
Function Det(b: mt; m,r,s: byte): real;
Var p,q: byte; d: real;
Begin
 For p:=r to m do
   For q:=1 to m+1 do b[p,q]:=b[p+1,q];
 For p:=1 to m do
   For q:=s to m do b[p,q]:=b[p,q+1];
 d:=0;
 If m=1 then d:=b[1,1]
   Else
      For p:=1 to m do d:=d-sign(p)*b[p,1]*det(b,m-1,p,1);
 Det:=d;
End;

BEGIN
  Clrscr;
  Nhap;
  d:=det(a,n,n+1,n+1); {detA}
  For i:=1 to n do
    For j:=1 to n do c[i,j]:=sign(i+j)*det(a,n-1,j,i)/d;
  Xuat;
END.
```

17. Program P30117; {Duong gap khuc Con Rong}

```
Uses Crt,Graph;
Var n,Gd,Gm: integer;
Procedure Conrong(x1,y1,x2,y2,m: integer);
Var x,y,x0,y0: integer;
Begin
 Randomize;
 If Random(2)=0 then
    begin
         x:=(x1+x2) div 2 +(y1-y2) div 2;
         y:=(y1+y2) div 2 +(x2-x1) div 2;
    end
```

```
  Else
     begin
          x:=(x1+x2) div 2 +(y2-y1) div 2;
          y:=(y1+y2) div 2 +(x1-x2) div 2;
     end;
 If m=1 then Line(x1,y1,x,y) Else Conrong(x1,y1,x,y,m-1);
 If m=1 then Line(x,y,x2,y2) Else Conrong(x2,y2,x,y,m-1);
End;
BEGIN
  Clrscr;
  Write(' Nhap N: '); Readln(n);
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  Conrong(200,400,500,200,n);
  OutTextXY(10,10,'Xong. Nhan Enter de thoat...');
  Readln;
  CloseGraph;
END.
```

Bài 2. Vào/Ra file

1. a, c, d, e: đúng.

   b: sai.

3. Program P30203; {Ma tran.ke}

```
Uses Crt;
Const   fi='P30203.inp';
        fo='P30203.out';
Var   a: array [1..200,1..200] of byte;
      n: byte;
      f: text;

Procedure Input;
Var i,j: byte;
Begin
 Assign(f,fi); Reset(f);
 Readln(f,n);
 For i:=1 to n do
   begin
       For j:=1 to n do Read(f,a[i,j]); Readln(f);
   end;
 Close(f);
End;

Function Test: boolean; {kiem tra do thi vo huong}
Var i,j: byte;
Begin
 test:=false;
 For i:=1 to n-1 do
   For j:=i+1 to n do
     If a[i,j]<>a[j,i] then exit;
 test:=true;
End;

Procedure Output;
Var i,j: byte;
Begin
 Assign(f,fo); Rewrite(f);
 Writeln(f,n);
 For i:=1 to n do
   begin
```

```
      For j:=1 to n do
        If (a[i,j]=1)and(i<>j) then Write(f,' ',j);
      Writeln(f);
  end;
 Close(f);
End;

BEGIN
 Input;
 Output;

End.
```

5. Program P30205; {kiem tra file doi xung}

```
Uses Crt;
Const  fi = 'P30205.TXT';
Var   f: file of char;
      c1,c2: char;
      size,d1,d2: word;

BEGIN
 Assign(f,fi); Reset(f);
 d2:=FileSize(f)-1;

 d1:=0;
 While (abs(d1-d2)>1) do
   begin
        Seek(f,d1); Read(f,c1);
        Seek(f,d2); Read(f,c2);
        Inc(d1); Dec(d2);
        If c1<>c2 then
          begin
               Write(' Day khong phai file doi xung');
               Close(f);
               Readln;
               Exit;
          end;
   end;
 Write(' Day la file doi xung');
 Readln;
 Close(f)
END.
```

7. Program P30207; {Sua van ban}

```
Uses Crt;
Const   fi='P30206.TXT';
Var   f: text;
      s: string;
      ch: char;

BEGIN
 Assign(f,fi); Reset(f);
 s:='';
 Read(f,ch);
   If (ch= '#')then Delete(s,length(s),1) Else s:=s+ch;
 Until Eof(f);
 Close(f);
 Assign(f,fi); Rewrite(f);
 Writeln(f,s);
 Close(f);
END.
```

9. Program P30209; {Tron 2 file}

```
Uses Crt;
Const   f1 = 'file1.txt';
        f2 = 'file2.txt';
        f3 = 'file3.txt';
Var   f,g,h: text;
      a,b: integer;
      Fa,Fb: boolean;

Procedure Doc(Var FN: text; Var VN: integer; Var VF: boolean);
Begin
 If not SeekEof(FN) then Read(FN,VN) Else VF:=true;
End;

Procedure Ghi(VN: integer);
Begin
 Write(h,' ',VN);
End;

Procedure Ghi2(Var FN: text; VN: integer);
Begin
 Write(h,' ',VN);
 While not SeekEof(FN) do
```

```
    begin
        Read(FN,VN);
        Write(h,' ',VN);
    end;
End;

BEGIN
 Assign(f,f1); Reset(f);
 Assign(g,f2); Reset(g);
 Assign(h,f3); Rewrite(h);
 Fa:=false; Fb:=false;
 Doc(f,a,Fa); Doc(g,b,Fb);
 While not(Fa or Fb) do
   If(a<b) then begin Ghi(a); Doc(f,a,Fa); end
      Else begin Ghi(b); Doc(g,b,Fb); end;
 If Fa then Ghi2(g,b);
 If Fb then Ghi2(f,a);
 Close(h); Close(f); Close(g);
END.
```

13. Program P30213; {Ghi file Record theo thu tu nguoc lai}

```
Uses Crt;
Const   fi = 'text.inp';
        fo = 'text.out';
Type   rec= Record
            d: integer;
          End;
Var   f: File of rec;
      buf: array [1..10000] of rec;
      n,i: integer;

BEGIN
 n:=0;
 Assign(f,fi); Reset(f);
 Repeat
   Inc(n);
   Read(f,Buf[n]);
 Until Eof(f);
 Assign(f,fo); Rewrite(f);
 For i:=n downto 1 do Write(f,Buf[i]);
```

```
 Close(f);
END.
```

15. Program P30215; {Dem tu}

```
Uses Crt;
Const   fi = 'hoten.txt';
Var   f: text;
       s: string;
      dem: word;

BEGIN
 Clrscr;
 dem:=0;
 Assign(f,fi); Reset(f);
 While not Eof(f) do
   begin
        Readln(f,s);
        While s[1]=' ' do Delete(s,1,1);
        While length(s)>0 do
  Begin
     Case s[1] of
        ' ': While (s[1]=' ')and(length(s)>0) do Delete(s,1,1);
          Else begin
                 inc(dem);
                  While (s[1]<>' ')and(length(s)>0) do Delete(s,1,1);
               end;
   End;
    end;
    end;
 Close(f);
 Write(' So tu la: ',dem);
 Readln;
END.
```

17. Program P30217; {Cat khoang trong thua}

```
Uses Crt;
Const   fi = 'P30217.inp';
        fo = 'P30217.out';
Var   inp,out: text;
      s: string;
```

```
BEGIN
 Assign(inp,fi); Reset(inp);
 Assign(out,fo); Rewrite(out);
 While not Eof(inp) do
    begin
          Readln(inp,s);
          While (s[1]=' ')and(Length(s)>0) do Delete(s,1,1);
          While (s[Length(s)]=' ')and(Length(s)>0) do
Delete(s,Length(s),1);
          While (Length(s)>0)and(pos(' ',s)<>0) do Delete(s,pos(' ',s),1);
          Writeln(out,s);
    end;
 Close(out); Close(inp);
END.
```

Bài 3. Số nguyên và hệ đếm

```
1. Program P30301;
{Tinh so chu so khac nhau trong bieu dien thap phan cua M/N}
Uses Crt;
Var   m,n,t,k: LongInt;
      cs: set of 0..9;
      r: array [1..MaxInt] of 0..9; {Mang so du}

Function Trung: Boolean;
Var i,kt: LongInt;
Begin
 kt:=0;
 For i:=1 to k-1 do If r[k]=r[i] then Inc(kt);
 If kt>0 then Trung:=true Else Trung:=false;
End;

Function S: byte;
Var i,kt: byte;
Begin
 kt:=0;
 For i:=0 to 9 do  If i in cs then Inc(kt);
 S:=kt;
End;

BEGIN
 ClrScr;
 Write('Nhap so M: '); Readln(m);
 Write('Nhap so N: '); Readln(n);
 t:=m div n;
 Write('M/N = ',t,'.');
 Repeat
   cs:=cs+[t mod 10];
   t:=t div 10
 Until t=0;
 m:=m mod n;
 k:=0;
 While (m<>0)and(s<10)and(not Trung) do
   begin
        m:=m*10;
```

```
        t:=m div n;
        cs:=cs+[t];
        Write(t);
        m:=m mod n;
        Inc(k);
        r[k]:=m;
    end;
  Writeln;
  Write('Trong bieu dien thap phan cua M/N co ',s,' chu so khac nhau.');
  Readln;
END.
```

5. Program P30305; { Chuyen tu he nhi phan sang he oct }

```
Uses Crt;
Const   oct: array [0..7] of char = ('0','1','2','3','4','5','6','7');
Var  s1,s2: String;
     i: byte;
     t: longint;

BEGIN
  Clrscr;
  Write('Nhap so nhi phan T: '); Readln(s1);
  t:=0;
  For i:=1 to length(s1) do
    If s1[i]='1' then t:=t Shl 1+1
      Else t:=t Shl 1;
  s2:='';
  Repeat
    s2:=oct[t mod 8]+s2;
    t:=t div 8;
  Until t=0;
  Writeln('So T trong he oct la: ',s2);
  Readln;
END.
```

7. Program P30307; {Kim gio kim phut}

```
Uses Crt;
Var   h,m,s: integer;
      d,hv,hp,mp,hr: real;
```

```
BEGIN
 Clrscr;
 Writeln('Thoi gian hien tai la h:m:s');
 Write(' Nhap h: '); Readln(h);
 Write(' Nhap m: '); Readln(m);
 Write(' Nhap s: '); Readln(s);
 hv:=2*pi/12; {van toc kim gio}
 hp:=(h+m/60+s/3600)*hv; {vi tri kim gio}
 mp:=(m+s/60)*(12*hv)/60; {vi tri kim phut}
 If hp>=mp then d:=hp-mp {Khoang cach tu}
  Else d:=2*pi+hp-mp; {kim phut den kim gio}
 hr:=d/(11*hv); {thoi gian kim phut duoi kip kim gio}
 Write('Thoi diem kim phut gap kim gio la: ');
 hr:=h+hr+m/60+s/3600;
 Write(int(hr):2:0,':');
 hr:=(hr-int(hr))*60;
 Write(int(hr):2:0,':');
 hr:=(hr-int(hr))*60;
 Write(int(hr):2:0);
 Readln;
END.
```

9. Program P30309; {Viet so bang chu so La Ma}

```
Uses Crt;
Const   ch: array [1..7] of char = ('I','V','X','S','C','D','M');
        vt: array [1..3] of byte = (2,4,6);
        d : array [1..9,1..2] of shortint = ((1,1),(1,2),(1,3),(2,-
1),(2,0),(2,1),(2,2),(2,3),(3,-1));
Var   i,j: byte;
      s,ch1,ch2: string;
      n,k: integer;

BEGIN
ClrScr;
s:='';
Write(' Nhap N: '); Readln(n);
i:=0;
While N>0 do
   begin
        k:=n mod 10;
        n:=n div 10;
        If k<>0 then
```

```
        begin
            If d[k,1]<>1 then ch1:=ch[d[k,1]+i] Else ch1:='';
            ch2:='';
            For j:=1 to abs(d[k,2]) do ch2:=ch2+ch[i+1];
            If d[k,2]>0 then s:=ch1+ch2+s Else s:=ch2+ch1+s;
        end;
      Inc(i,2);
  end;
Write(s);
Readln;
END.
{ I = 1
 V = 5
 X = 10
 S = 50
 C = 100
 D = 500
 M = 1000}
```

11. Program P30311; {Giai pt nguyen ax+by=d voi d=USCLN(a,b)}

```
Uses Crt; {thuat toan Euclide}
Var  a,aa,b,bb,p,q,r,s,pp,qq,tg: integer;

BEGIN
 Clrscr;
 Write(' Nhap a: '); Readln(a);
 Write(' Nhap b: '); Readln(b);
 aa:=a; bb:=b;
 p:=0; q:=1;
 pp:=1; qq:=0;
 r:=a mod b;
 While r<>0 do
   begin
        s:=a div b;
        a:=b; b:=r;
        tg:=pp; pp:=p; p:=tg-s*p;
        tg:=qq; qq:=q; q:=tg-s*q;
        r:=a mod b;
   end;
 Write(p,'*',aa,'+',q,'*',bb,' = ',b);
 Readln;
END.
```

```
12. Program P30312; {Tinh cac so lon}
Uses Crt;
Type solon = array[0..255] of byte;

Procedure chen(t,d: integer; Var s: solon);
Var i: integer;
Begin
  Inc(s[0]);
  For i:=s[0] downto d+1 do s[i]:=s[i-1];
  s[d]:=t;
End;

Procedure cut(d: integer; Var s: solon);
Var i: integer;
Begin
  Dec(s[0]);
  For i:=d to s[0] do s[i]:=s[i+1];
End;

Procedure nhap(Var s: solon);
Var str: String;
       i: integer;
Begin
  Readln(str);
  s[0]:=Length(str);
  For i:=Length(str) downto 1 do s[s[0]-i+1]:=ord(str[i])-48;
End;

Procedure hien(s: solon);
Var i: integer;
Begin
  For i:=s[0] downto 1 do Write(s[i]);
End;

Function max(a,b: integer): integer;
Begin
  If a<b then max:=b Else max:=a;
End;

Procedure cong(Var s1,s2: solon);
```

```
Var i,nho: integer;
Begin
 nho:=0;
 For i:=1 to max(s1[0],s2[0]) do
   begin
        nho:=nho+s1[i]+s2[i];
        s1[i]:=nho mod 10;
        nho:=nho div 10;
   end;
 s1[0]:=i;
 If nho>0 then
    begin
         Inc(s1[0]);
         s1[s1[0]]:=nho;
     end;
 End;

 Procedure tru(Var s1,s2: solon);
 Var nho,i: Integer;
 Begin
  nho:=0;
  For i:=1 to s2[0] do
    If s1[i]<s2[i]+nho then
         begin
              s1[i]:=10+s1[i]-nho-s2[i];
              nho:=1;
          end
       Else
        begin
             s1[i]:=s1[i]-s2[i]-nho;
             nho:=0;
         end;
   While (nho>0)and(i<255) do
    begin
          Inc(i);
          If s1[i]=0 then s1[i]:=9
           Else
```

```
            begin
                dec(s1[i]);
                nho:=0;
            end;
    end;
  While (s1[s1[0]]=0)and(s1[0]>0) do Dec(s1[0]);
End;

Procedure nhannho(Var s: solon; t: integer);
Var  nho,i: word;
Begin
 nho:=0;
 For i:=1 to s[0] do
   begin
        nho:=nho+s[i]*t;
        s[i]:=nho mod 10;
        nho:=nho div 10;
   end;
 While nho>0 do
   begin
        Inc(s[0]);
        s[s[0]]:=nho mod 10;
        nho:=nho div 10;
   end;
End;

Procedure nhan(Var s1,s2: solon);
Var c,d: solon;
     i,j,t: integer;
Begin
 FillChar(c,SizeOf(c),0);
 For i:=1 to s2[0] do
   begin
         d:=s1;
         nhannho(d,s2[i]);
         For j:=2 to i do nhannho(d,10);
         cong(c,d);
    end;
 s1:=c;
End;
```

```
Function sosanh(Var s1,s2: solon): shortint;
Var i: integer;
Begin
 While(s1[s1[0]]=0)and(s1[0]>0) do dec(s1[0]);
 While(s2[s2[0]]=0)and(s2[0]>0) do dec(s2[0]);
 If s1[0]>s2[0] then begin sosanh:=1; exit; end;
 If s1[0]<s2[0] then begin sosanh:=-1; exit; end;
 For i:=s1[0] downto 1 do
   begin
        If s1[i]>s2[i] then begin sosanh:=1; exit; end;
        If s1[i]<s2[i] then begin sosanh:=-1; exit; end;
   end;
 sosanh:=0;
End;

Procedure laydu(Var s1,s2: solon; Var t: integer);
Begin
 t:=0;
 While sosanh(s1,s2)>=0 do
   begin  tru(s1,s2); Inc(t);  end;
End;

Procedure chia(Var s1,s2,s3: solon);
Var   s,s5,s6,s4: solon;
      a,b: byte;
Begin
 FillChar(s,SizeOf(s),0);
 While sosanh(s1,s2)>=0 do
   begin
        s4:=s2; a:=0;
        While (sosanh(s1,s4)>=0) do  begin chen(0,1,s4); Inc(a); end;
        cut(1,s4); s5:=s4; b:=0;
        While (sosanh(s1,s5)>=0) do begin  cong(s5,s4); Inc(b); end;
        FillChar(s6,SizeOf(s6),0);
        s6[0]:=a; s6[s6[0]]:=b;
        cong(s,s6); nhan(s6,s2); tru(s1,s6);
   end;
 While(s1[s1[0]]=0)and(s1[0]>0) do dec(s1[0]);
 While(s[s[0]]=0)and(s[0]>0) do dec(s[0]);
 s3:=s1; s1:=s;
End;
```

```
Var s1,s2,s3: solon;

BEGIN
 {Chuong trinh sau chi thuc hien phep chia}
 Clrscr;
 Write('Nhap so thu nhat: '); Nhap(s1);
 Write('Nhap so thu hai : '); Nhap(s2);
 Writeln('Ket qua sau khi thuc hien phep chia:');
 chia(s1,s2,s3);
 Write('Thuong: ');
 hien(s1);
 Writeln;
 Write('So du: ');
 hien(s3);
 Readln;
END.
```

13.
```
Program P30313; {Tinh 2^200}
{$I P30312.pas}
Var  i: byte;
     s: solon;

BEGIN
 Clrscr;
 FillChar(s,Sizeof(s),1);
 s[0]:=1;
 For i:=1 to 200 do nhannho(s,2);
 Writeln(' 2^200 la: ');
 Hien(s);
 Readln;
END.
```

15.
```
{$I P30312.pas}
Program P30315; {Doi so thap phan lon thanh nhi phan}
Var  s,s1,s3: solon;
     d: Array [1..maxint] of byte;
     n,i: integer;
```

```
BEGIN
 Clrscr;
 Write(' Nhap so lon N = '); Nhap(s);
 s1[0]:=1; s1[1]:=2; n:=0;
 Repeat
   chia(s,s1,s3);
   Inc(n);
   d[n]:=s3[1];
 Until s[0]=0;
 For i:=n downto 1 do Write(d[i]);
 Readln;
END.
```

17. Program P30317;{tinh toan voi xau nhi phan}

```
{$I P30312.pas}
Type  nhiphan=solon;

Procedure haimu(n: byte; Var s: solon);
Var i: byte;
Begin
  Fillchar(s,sizeof(s),0);
  s[0]:=1; s[1]:=1;
  For i:=1 to n do nhannho(s,2);
End;
Procedure dectobin(s: solon; Var n: nhiphan);
Var s1,s3: solon;
Begin
  s1[0]:=1; s1[1]:=2;
  FillChar(n,SizeOf(n),0);
  Repeat
     chia(s,s1,s3);
     Inc(n[0]);
     n[n[0]]:=s3[1];
  Until s[0]=0;
End;
Procedure bintodec(n: nhiphan; Var s: solon);
Var s1: solon; i: byte;
Begin
  Fillchar(s,sizeof(s),0);
  s[0]:=1;
  For i:=1 to n[0] do
    If n[i]=1 then
```

```
    begin
        haimu(i-1,s1);
        cong(s,s1);
    end;
End;
Procedure nhanbin(Var n1,n2: nhiphan);
Var s1,s2: solon;
Begin
 bintodec(n1,s1); bintodec(n2,s2);
 nhan(s1,s2);
 dectobin(s1,n1);
End;

Procedure trubin(Var n1,n2: nhiphan);
Var s1,s2: solon;
Begin
 bintodec(n1,s1); bintodec(n2,s2);
 tru(s1,s2);
 dectobin(s1,n1);
End;

Procedure congbin(Var n1,n2: nhiphan);
Var s1,s2: solon;
Begin
 bintodec(n1,s1); bintodec(n2,s2);
 cong(s1,s2);
 dectobin(s1,n1);
End;
Procedure chiabin(Var n1,n2,n3: nhiphan);
Var s1,s2,s3: solon;
Begin
 bintodec(n1,s1); bintodec(n2,s2);
 chia(s1,s2,s3);
 dectobin(s1,n1); dectobin(s3,n3);
End;
```

Bài 4. Đồ hoạ

1. Program P30401; {Tam giac dac deu quay quanh tam voi van toc deu}

```
Uses Crt,Graph;
Const   lech=0.06;
        kc=120;
        xt=320; yt=240;
        mau=1;
        tre=80;
Var Gd,Gm: integer;
    d: real;
Procedure Ve;
Var x1,x2,x3,y1,y2,y3: integer;
Begin
 x1:=xt+round(kc*sin(d)); y1:=yt+round(kc*cos(d));
 x2:=xt+round(kc*sin(d+2*pi/3)); y2:=yt+round(kc*cos(d+2*pi/3));
 x3:=xt+round(kc*sin(d+4*pi/3)); y3:=yt+round(kc*cos(d+4*pi/3));
 Line(x1,y1,x2,y2);
 Line(x2,y2,x3,y3);
 Line(x3,y3,x1,y1);
 FloodFill(xt,yt,mau);
End;

BEGIN
 Gd:=Detect;
 InitGraph(Gd,Gm,' ');
 SetColor(mau);
 SetFillStyle(1,mau);
 d:=0;
 Repeat
  Delay(tre);
  d:=d+lech;
  ClearDevice;
  Ve;
 Until KeyPressed;
 CloseGraph;
End.
```

3. Program P30403; {Dao dong dieu hoa tren mot doan nam ngang}

```
Uses Crt,Graph;
Const   c=300; w=10; tg=0.001; r=2; tre=4; y=240;
```

```
Var Gd,Gm,x: integer;
     t: real;
     p: pointer;
BEGIN
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  x:=320-(c div 2)+Round(c*(1+cos(w*t))/2);
  SetColor(1);
  Circle(x,y,r);
  SetFillStyle(1,1);
  FloodFill(x,y,1);
  Getmem(p,ImageSize(x-r-5,y-r,x+r+5,y+r));
  Getimage(x-r-5,y-r,x+r+5,y+r,p^);
  t:=0;
  Repeat
    Putimage(x-r-5,y-r,p^,normalput);
    Delay(tre);
    x:=320-(c div 2)+Round(c*(1+cos(w*t))/2);
    t:=t+tg;
  Until KeyPressed;
  Closegraph;
END.
```

5. Program P30405;

```
{Hien thi cac hinh tron chuyen dong xien voi van toc khac nhau}
Uses Crt,Graph;
Const   BanKinh=15; Smax=8; Tre=10; Nmax=5;
Var  n: byte;
     Gd,Gm: integer;
     x,y: array [1..Nmax] of integer;
     Huong: array [1..Nmax] of real;
     MauBong,s: array [1..Nmax] of byte;

Procedure KhoiTao;
Var i: byte;
Begin
Randomize;
n:=Random(Nmax-2)+3;
For i:=1 to n do
  begin
       x[i]:=Random(640-2*BanKinh)+BanKinh;
       y[i]:=Random(480-2*BanKinh)+BanKinh;
```

```
        MauBong[i]:=Random(15)+1;
        SetColor(MauBong[i]);
        Circle(x[i],y[i],BanKinh);
        SetFillStyle(1,MauBong[i]);
        FloodFill(x[i].y[i],MauBong[i]);
        Huong[i]:=Random(360)*pi/180;
        s[i]:=Random(Smax)+1;
   end;
End;
Procedure ChuyenDong;
Var i: byte;
Begin
For i:=1 to n do
   begin
        x[i]:=x[i]+round(s[i]*cos(Huong[i]));
        y[i]:=y[i]+round(s[i]*sin(Huong[i]));
        SetColor(MauBong[i]);
        Circle(x[i],y[i],BanKinh);
        SetFillStyle(1,MauBong[i]);
        FloodFill(x[i],y[i],MauBong[i]);
        If x[i]<=BanKinh then
          If Huong[i]>pi then Huong[i]:=3*pi-Huong[i]
            Else Huong[i]:=pi-Huong[i];
        If x[i]>=640-BanKinh then
          If Huong[i]<pi then Huong[i]:=pi-Huong[i]
            Else Huong[i]:=3*pi-Huong[i];
        If y[i]<=BanKinh then Huong[i]:=2*pi-Huong[i];
        If y[i]>=480-BanKinh then Huong[i]:=2*pi-Huong[i];
   end;
End;
Begin
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  KhoiTao;
  While not KeyPressed do
    begin
        Delay(Tre);
        ClearDevice;
        ChuyenDong;
    end;
  CloseGraph;
End.
```

9. Program P30409; {HideMouse}

```
Uses   Crt,Dos,Graph;
Var   reg: registers;
      XMouse,YMouse,Gd,Gm,x,y: integer;
      S1,S2: string[11];
Procedure InitMouse;
Begin
 reg.ax:=$0000;
 intr($33,reg);
End;

Procedure ShowMouse;
Begin
 reg.ax:=$0001;
 intr($33,reg);
End;
Procedure HideMouse;
Begin
 reg.ax:=$0002;
 intr($33,reg);
End;

Procedure GetMousePosition(Var Xmouse,YMouse: integer);
Begin
 reg.ax:=$0003;
 intr($33,reg);
 XMouse:=reg.cx; YMouse:=reg.dx;
End;

BEGIN
 Gd:=Detect;
 InitGraph(Gd,Gm,' ');
 InitMouse;
 ShowMouse;
 Repeat
  GetMousePosition(XMouse,YMouse);
  If (XMouse<>x)or(YMouse<>y) then
    begin
        HideMouse;
        Setcolor(16);
        Str(x,S1); Str(y,S2);
        Outtextxy(x,y, S1+':'+S2);
```

```
        Setcolor(15);
        Str(Xmouse,S1); Str(Ymouse,S2);
        Outtextxy(XMouse,YMouse, S1+':'+S2);
        x:=XMouse; y:=YMouse;
        ShowMouse;
    end;
 Until KeyPressed;
 CloseGraph;
END.
```

11. Program P30411; {Cac hinh vuong doi mau}

```
Uses Crt,Dos,Graph;
Const   w=20; x0=200; y0=100;
Var  reg: registers;
     XMouse,YMouse,Gd,Gm,i,j: integer;
     a: array [0..9, 0..9] of byte;

Procedure InitMouse;
Begin
 reg.ax:=$0000;
 intr($33,reg);
End;

Procedure ShowMouse;
Begin
 reg.ax:=$0001;
 intr($33,reg);
End;

Procedure HideMouse;
Begin
 reg.ax:=$0002;
 intr($33,reg);
End;

Function MouseClick: char;
Begin
 reg.ax:=$0003;
 intr($33,reg);
 If reg.bx=1 then MouseClick:='L';
 If reg.bx=2 then MouseClick:='R';
End;
```

```
Procedure GetMousePosition(Var XMouse,YMouse: integer);
Begin
 reg.ax:=$0003;
 intr($33,reg);
 XMouse:=reg.cx;
 YMouse:=reg.dx;
End;

BEGIN
 Gd:=Detect;
 InitGraph(Gd,Gm,' ');
 SetFillStyle(1,11);
 Bar(x0,y0,x0+10*w-1,y0+10*w-1);
 InitMouse;
 ShowMouse;
 Repeat
   GetMousePosition(XMouse,YMouse);
   If (XMouse>=x0)and(XMouse<x0+10*w)and(YMouse>=y0)
         and(YMouse<y0+10*w) and(MouseClick='L') then
       begin
            j:=(XMouse-x0)div w; i:=(YMouse-y0)div w;
            If a[i,j]<>4 then a[i,j]:=4 Else a[i,j]:=11;
            SetFillStyle(1,a[i,j]);
            HideMouse;
            Bar(x0+j*w,y0+i*w,x0+(j+1)*w-1,y0+(i+1)*w-1);
            ShowMouse;
            Delay(300);
       end;
 Until MouseClick='R';
 CloseGraph;
END.
```

13. Program P30413; {Bang chu nhat}

```
Uses Crt,Dos,Graph;
Var Gd,Gm: integer;
     reg: registers;
     a: array [1..2,1..3,1..2] of integer;

Procedure InitMouse;
Begin
```

```
 reg.ax:=$0000;
 intr($33,reg);
End;

Procedure ShowMouse;
Begin
 reg.ax:=$0001;
 intr($33,reg);
End;

Procedure GetMousePosition(Var xmouse,ymouse: integer);
Begin
 reg.ax:=$0003;
 intr($33,reg);
 xmouse:=reg.cx;
 ymouse:=reg.dx;
End;

Procedure HideMouse;
Begin
 reg.ax:=$0002;
 intr($33,reg);
End;

Procedure Ve_bang;
Var i,j: byte; xd,yd: integer;
Begin
 SetColor(7);
 xd:=240; yd:=190;
 For i:=0 to 1 do
  begin
      Line(xd,yd+i*50,xd+150,yd+i*50);
      For j:=0 to 2 do
        begin
            Line(xd+j*50,yd,xd+j*50,yd+100);
            a[i+1,j+1,1]:=xd+50*j;
            a[i+1,j+1,2]:=yd+50*i;
        end;
  end;
 Line(xd,yd+100,xd+150,yd+100); Line(xd+150,yd,xd+150,yd+100);
 SetFillStyle(1,1); FloodFill(a[1,1,1]+1,a[1,1,2]+1,7);
```

```
 SetFillStyle(1,2); FloodFill(a[1,2,1]+1,a[1,2,2]+1,7);
 Setfillstyle(1,3); FloodFill(a[1,3,1]+1,a[1,3,2]+1,7);
 SetFillStyle(1,4); FloodFill(a[2,1,1]+1,a[2,1,2]+1,7);
 SetFillStyle(1,15); FloodFill(a[2,2,1]+1,a[2,2,2]+1,7);
 SetFillStyle(1,5); FloodFill(a[2,3,1]+1,a[2,3,2]+1,7);
End;

Procedure Tim(Var xa,ya,xt,yt: integer; m: byte);
Var i,j: byte;
Begin
 xt:=0;
 For i:=1 to 2 do
  For j:=1 to 3 do
   If Getpixel(a[i,j,1]+1,a[i,j,2]+1)=m then
     begin
          xa:=a[i,j,1]+1; ya:=a[i,j.2]+1;
          If (i>1)and(getpixel(a[i-1,j,1]+1,a[i-1,j,2]+1)=15) then
               begin xt:=a[i-1,j,1]+1; yt:=a[i-1,j,2]+1; exit; end;
          If (i<2)and(getpixel(a[i+1,j,1]+1,a[i+1,j,2]+1)=15) then
               begin xt:=a[i+1,j,1]+1; yt:=a[i+1,j,2]+1; exit; end;
          If (j>1)and(getpixel(a[i,j-1,1]+1,a[i,j-1,2]+1)=15) then
               begin xt:=a[i,j-1,1]+1; yt:=a[i,j-1,2]+1; exit; end;
          If (j<3)and(getpixel(a[i,j+1,1]+1,a[i,j+1,2]+1)=15) then
                begin xt:=a[i,j+1,1]+1; yt:=a[i,j+1,2]+1; exit; end;
          exit;
     end;
End;

Procedure Doi;
Var mau,xa,ya,xt,yt,xm,ym: integer;
Begin
 GetMousePosition(xm,ym);
 HideMouse;
 mau:=Getpixel(xm,ym);
 ShowMouse;
 If (mau=0)or(mau=7) then exit;
 Tim(xa,ya,xt,yt,mau);
 If xt=0 then exit;
 HideMouse;
 SetColor(mau);
 SetFillStyle(1,mau);
 FloodFill(xt,yt,7);
```

```
 SetFillStyle(1,15);
 FloodFill(xa,ya,7);
 ShowMouse;
End;

BEGIN
 Gd:=Detect;
 InitGraph(Gd,Gm,'');
 Ve_bang;
 InitMouse;
 ShowMouse;
 While true do
   begin
       reg.ax:=$0003;
       intr($33,reg);
       If (reg.bx and 1)=1 then Doi;
       If (reg.bx and 2)=2 then break;
   end;
 CloseGraph;
END.
```

15. Program P30415;

```
{Hien thi con Xe va con Hau tren ban co va dieu khien no}
Uses Crt,Dos,Graph;
Type   ba= array [0..0]of byte;
       bap=^ba;
Var   reg: registers;
      Gd,Gm: integer;
      ChonXe,ChonHau: boolean;
      xHau,yHau,xHauL,yHauL,xXe,yXe,xXeL,yXeL: integer;
      xMouse,yMouse: integer;
      a: array [1..8,1..8,1..2] of integer;
      b: array [1..8,1..8] of byte;
      ODen,OTrang,HauTrang,HauDen,XeTrang,XeDen: bap;

Procedure InitMouse;
Begin
 reg.ax:=$0000;
 intr($33,reg);
End;

Procedure ShowMouse;
```

```
Begin
 reg.ax:=$0001;
 intr($33,reg);
End;

Procedure GetMousePosition(Var xmouse,ymouse:integer);
Begin
 reg.ax:=$0003;
 intr($33,reg);
 xmouse:=reg.cx; ymouse:=reg.dx;
End;

Procedure HideMouse;
Begin
 reg.ax:=$0002;
 intr($33,reg);
End;

Function MouseClick: char;
Var n: char;
Begin
 reg.ax:=$0003;
 intr($33,reg);
 n:='A';
 If (reg.bx and 1)=1 then n:='L';
 If (reg.bx and 2)=2 then n:='R';
 MouseClick:=n;
End;

Procedure CatHinh;
Begin
 SetColor(14);
 Rectangle(1,1,51,51);
 SetFillStyle(1,8); FloodFill(2,2,14);
 Getmem(ODen,ImageSize(1,1,51,51));
 GetImage(1,1,51,51,ODen^);
 SetFillStyle(1,15); FloodFill(2,2,14);
 Getmem(OTrang,ImageSize(1,1,51,51));
 Getimage(1,1,51,51,OTrang^);
 SetColor(2);
 SetTextStyle(0,0,3);
```

```
  Circle(26,26,20);
  SetFillStyle(1,4); FloodFill(26,26,2);
  OutTextxy(16,16,'H');
  Getmem(HauTrang,ImageSize(1,1,51,51));
  Getimage(1,1,51,51,HauTrang^);
  SetColor(4); OutTextxy(16,16,'H');
  SetColor(2); OutTextxy(16,16,'X');
  Getmem(XeTrang,imagesize(1,1,51,51));
  Getimage(1,1,51,51,XeTrang^);
  Putimage(1,1,ODen^,normalput);
  Circle(26,26,20);
  SetFillStyle(1,4); FloodFill(26,26,2);
  OutTextxy(16,16,'H');
  Getmem(HauDen,ImageSize(1,1,51,51));
  Getimage(1,1,51,51,HauDen^);
  SetColor(4); OutTextxy(16,16,'H');
  SetColor(2); OutTextxy(16,16,'X');
  Getmem(XeDen,imagesize(1,1,51,51));
  Getimage(1,1,51,51,XeDen^);
End;

Procedure VeBang;
Var xd,yd,l,i,j: integer;
Begin
  ClearDevice;
  SetTextStyle(1,0,1);
  OutTextxy(10,10,'Qufn hfu : H - Qufn Xe : X');
  l:=50;
  xd:=(640-8*l)div 2; yd:=(480-8*l)div 2;
  SetColor(14);
  Rectangle(xd,yd,xd+l*8,yd+l*8);
  For i:=1 to 7 do
     begin
          Line(xd+l*i,yd,xd+l*i,yd+l*8);
          Line(xd,yd+l*i,xd+l*8,yd+l*i);
       end;
  For i:=1 to 8 do
    For j:=1 to 8 do
      begin
          a[i,j,1]:=xd+(j-1)*l;
```

```
        a[i,j,2]:=yd+(i-1)*l;
        If (i+j)mod 2=0 then SetFillStyle(1,8)
          Else SetFillStyle(1,15);
        FloodFill(a[i,j,1]+1,a[i,j,2]+1,14);
    end;
ChonHau:=false; ChonXe:=false;
Randomize;
xHauL:=Random(8)+1; yHauL:=Random(8)+1;
Repeat
   xXeL:=Random(8)+1;
   yXeL:=Random(8)+1;
 Until (xXeL<>xHauL)or(yXeL<>yHauL);
 If Odd(xHauL+yHauL) then
Putimage(a[xHauL,yHauL,1],a[xHauL,yHauL,2],HauTrang^,normalput)
   Else
Putimage(a[xHauL,yHauL,1],a[xHauL,yHauL,2],HauDen^,normalput);
 If Odd(xXeL+yXeL) then
Putimage(a[xXeL,yXeL,1],a[xXeL,yXeL,2],XeTrang^,normalput)
   Else
Putimage(a[xXeL,yXeL,1],a[xXeL,yXeL,2],XeDen^,normalput);
 FillChar(b,Sizeof(b),0);
 b[xHauL,yHauL]:=1; b[xXeL,yXeL]:=2;
End;

Procedure TimViTri(Var x,y: byte);
Var i,j: byte;
Begin
 x:=0;
 For i:=1 to 8 do
   For j:=1 to 8 do
    If (a[i,j,1]<xMouse)and(a[i,j,1]+49>xMouse)and
(a[i,j,2]<yMouse)and(a[i,j,2]+49>yMouse)  then
       begin
            x:=i; y:=j;
            exit;
       end;
End;

Function Ok: boolean;
Begin
  Ok:=true;
```

```
 If (xHau=xHauL)or(yHau=yHauL) then exit;
 If abs(xHau-xHauL)=abs(yHau-yHaul) then exit;
 Ok:=false;
End;

Procedure ChuyenHau(x,y: byte);
Begin
xHau:=x; yHau:=y;
If not Ok then exit;
HideMouse;
If Odd(xHauL+yHauL) then

Putimage(a[xHauL,yHauL,1],a[xHauL,yHauL,2],OTrang^,normalput)
  Else
Putimage(a[xHauL,yHauL,1],a[xHauL,yHauL,2],ODen^,normalput);
If Odd(xHau+yHau) then
      Putimage(a[xHau,yHau,1],a[xHau,yHau,2],HauTrang^,normalput)
  Else Putimage(a[xHau,yHau,1],a[xHau,yHau,2],HauDen^,normalput);
ShowMouse;
b[xHaul,yHauL]:=0;
b[xHau,yHau]:=1;
xHauL:=xHau; yHauL:=yHau;
End;

Procedure ChuyenXe(x,y: byte);
Begin
 xXe:=x; yXe:=y;
 If (xXe<>xXeL)and(yXe<>yXeL) then exit;
 HideMouse;
 If Odd(xXeL+yXeL) then
      Putimage(a[xXeL,yXeL,1],a[xXeL,yXeL,2],OTrang^,normalput)
   Else Putimage(a[xXeL,yXeL,1],a[xXeL,yXeL,2],ODen^,normalput);
 If Odd(xXe+yXe) then
      Putimage(a[xXe,yXe,1],a[xXe,yXe,2],XeTrang^,normalput)
   Else Putimage(a[xXe,yXe,1],a[xXe,yXe,2],XeDen^,normalput);
 ShowMouse;
 b[xXel,yXeL]:=0;
 b[xXe,yXe]:=2;
 xXeL:=xXe; yXeL:=yXe;
End;
Procedure ChuyenQuan;
Var x,y: byte;
```

```
Begin
 GetMousePosition(xMouse,yMouse);
 TimViTri(x,y);
 If x=0 then exit;
 If b[x,y]=1 then
   begin
        HideMouse;
        ChonHau:=true; ChonXe:=false;
        SetColor(0);
        OutTextxy(450,10,'Chon Xe');
        If Odd(xXeL+yXeL) then SetColor(15)
          Else SetColor(8);
        Circle(a[xXeL,yXeL,1]+25,a[xXeL,yXeL,2]+25,22);
        SetColor(1);
        OutTextxy(450,10,'Chon Hau');
        Circle(a[x,y,1]+25,a[x,y,2]+25,22);
        ShowMouse;
   end;
 If b[x,y]=2 then
    begin
         HideMouse;
         ChonHau:=false; ChonXe:=true;
         SetColor(0);
         OutTextxy(450,10,'Chon Hau');
         If Odd(xHauL+yHauL) then SetColor(15)
           Else SetColor(8);
         Circle(a[xHauL,yHauL,1]+25,a[xHauL,yHauL,2]+25,22);
         SetColor(1);
         OutTextxy(450,10,'Chon Xe');
         Circle(a[x,y,1]+25,a[x,y,2]+25,22);
         ShowMouse;
     end;
If (b[x,y]=0)and(not ChonHau)and(not ChonXe) then exit;
If b[x,y]=0 then
   begin
        If ChonHau then ChuyenHau(x,y);
        If ChonXe then ChuyenXe(x,y);
    end;
End;
BEGIN
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
```

```
 CatHinh;
 VeBang;
 InitMouse;
 ShowMouse;
 While true do
   begin
        reg.ax:=$0003;
        intr($33,reg);
        If (reg.bx and 1)=1 then ChuyenQuan;
        If (reg.bx and 2)=2 then break;
    end;
 CloseGraph;
END.
```

17. Program P30417; {Kiem tra viec re chuot}

```
Uses Crt,Dos,Graph;
Var   reg: registers;
      Gd,Gm: integer;
      xd,yd,xc,yc: integer;
      c: char;
      s1,s2,s3,s4: string;
Procedure InitMouse;
Begin
 reg.ax:=$0000;
 intr($33,reg);
End;

Procedure ShowMouse;
Begin
 reg.ax:=$0001;
 intr($33,reg);
End;

Procedure GetMousePosition(Var xmouse,ymouse: integer);
Begin
 reg.ax:=$0003;
 intr($33,reg);
 xmouse:=reg.cx;
 ymouse:=reg.dx;
End;
Procedure HideMouse;
Begin
```

```
 reg.ax:=$0002;
 intr($33,reg);
End;
Function MouseClick: char;
Var n: char;
Begin
 reg.ax:=$0003;
 intr($33,reg);
 n:='A';
 If (reg.bx and 1)=1 then n:='L';
 If (reg.bx and 2)=2 then n:='R';
 MouseClick:=n;
End;
BEGIN
 Gd:=Detect;
 InitGraph(Gd,Gm,' ');
 InitMouse;
 ShowMouse;
 SetColor(14);
 Repeat
    c:=MouseClick;
    If c='L' then
      begin
          ClearDevice;
          GetMousePosition(xd,yd);
          HideMouse;
          Str(xd,s1); Str(yd,s2);
          OutTextxy(xd,yd,s1+':'+s2);
          ShowMouse;
          While MouseClick='L' do;
          GetMousePosition(xc,yc);
          HideMouse;
          Str(xc,s3); Str(yc,s4);
          OutTextxy(xc,yc,s3+':'+s4);
          ShowMouse;
      end;
 Until c='R';
 CloseGraph;
END.
```

Bài 5. Đa thức

```
3. Program P30503;  {Tinh tich cua hai da thuc}
Uses Crt;
Type Dathuc = Record
            Deg : Byte;
            Koef: Array[0..100] of Real;
          End;
Var Dt1,Dt2,Dt3: Dathuc;

Procedure Nhapdathuc(Var Dt: Dathuc);
Var i: Byte;
Begin
 FillChar(Dt,SizeOf(Dt),0);
 With Dt do
   begin
        Write('Cho bac cua da thuc: '); Readln(Deg);
        For i:=Deg downto 0 do
          begin
               Write('Cho he so A[',i,'] cua da thuc: ');
               Readln(Koef[i]);
          end;
   end;
End;

Procedure Vietdathuc(Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
   begin
        Writeln('Bac cua da thuc la: ',Deg);
        For i:=Deg downto 0 do
        Writeln('He so A[',i,'] cua da thuc la: ',Koef[i]:10:2);
   end;
End;

Procedure Chuanhoa(Var Dt: Dathuc);
Var i: Byte;
Begin
 While Dt.Koef[Dt.Deg]=0 do Dt.Deg:=Dt.Deg-1;
End;
```

```
Procedure Tich(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 Dt3.Deg:=Dt1.Deg+Dt2.Deg;
 For i:=0 to Dt1.Deg do
   For j:=0 to Dt2.Deg do
Dt3.Koef[i+j]:=Dt3.Koef[i+j]+Dt1.Koef[i]*Dt2.Koef[j];
End;

BEGIN
 Clrscr;
 Writeln('Nhap da thuc thu nhat: ');
 Nhapdathuc(Dt1);
 Writeln('Nhap da thuc thu hai: ');
 Nhapdathuc(Dt2);
 Tich(Dt1,Dt2,Dt3);
 Writeln('Tich cua hai da thuc la: ');
 Vietdathuc(Dt3);
 Readln;
END.
```

5. Program P30505; {Chia hai da thuc}

```
Uses Crt;
Type Dathuc = Record
            Deg : Byte;
            Koef: Array [0..100] of Real;
          End;
Var  Dt1,Dt2,Dt3,Dt4: Dathuc;
     N: Byte;

Procedure Nhapdathuc(Var Dt: Dathuc);
Var i: Byte;
Begin
 FillChar(Dt,SizeOf(Dt),0);
 With Dt do
   begin
         Write('Cho bac cua da thuc: '); Readln(Deg);
         For i:=Deg downto 0 do
           begin
                Write('Cho he so A[',i,'] cua da thuc: ');
                Readln(Koef[i]);
```

```
            end;
    end;
End;

Procedure Vietdathuc(Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
    begin
          Writeln('Bac cua da thuc la: ',Deg);
          For i:=Deg downto 0 do Writeln('He so A[',i,'] cua da thuc la:
',Koef[i]:10:2);
    end;
End;

Procedure Chuanhoa(Var Dt: Dathuc);
Var i: Byte;
Begin
 While (Dt.Deg > 0)and(Dt.Koef[Dt.Deg] =0) do Dt.Deg:=Dt.Deg-1;
End;

Procedure Hieu(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i: Byte;
Begin
 Dt3.Deg:=Dt2.Deg;
 If Dt1.Deg > Dt2.Deg then Dt3.Deg:=Dt1.Deg;
 For i:=0 to Dt3.Deg do Dt3.Koef[i]:=Dt1.Koef[i]-Dt2.Koef[i];
 Chuanhoa(Dt3);
End;

Procedure Tich(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 Dt3.Deg:=Dt1.Deg+Dt2.Deg;
 For i:=0 to Dt1.Deg do
   For j:=0 to Dt2.Deg do
Dt3.Koef[i+j]:=Dt3.Koef[i+j]+Dt1.Koef[i]*Dt2.Koef[j];
End;

Procedure Chia(Dt1,Dt2: Dathuc; Var Dt3,Dt4: Dathuc);
Var Dt,Dt5: Dathuc; i,j: Byte;
```

```
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 If Dt1.Deg < Dt2.Deg then
   begin
        Dt4:=Dt1;
        Exit;
   end;
 Dt3.Deg:=Dt1.Deg-Dt2.Deg;
 Dt3.Koef[Dt1.Deg-Dt2.Deg]:=Dt1.Koef[Dt1.Deg]/Dt2.Koef[Dt2.Deg];
 Tich(Dt3,Dt2,Dt);
 Hieu(Dt1,Dt,Dt4);
 For i:=Dt1.Deg-1 downto Dt2.Deg do
    If Dt4.Deg >=Dt2.Deg then
       begin
            Dt3.Koef[i-
Dt2.Deg]:=Dt4.Koef[Dt4.Deg]/Dt2.Koef[Dt2.Deg];
            Fillchar(Dt5,SizeOf(Dt5),0);
            Dt5.Deg:=Dt4.Deg-Dt2.Deg;
            Dt5.Koef[Dt5.Deg]:=Dt3.Koef[i-Dt2.Deg];
            Tich(Dt5,Dt2,Dt);
            Hieu(Dt4,Dt,Dt4);
       end;
 End;

 BEGIN
  Clrscr;
  Writeln('Nhap da thuc thu nhat: ');
  Nhapdathuc(Dt1);
  Writeln('Nhap da thuc thu hai: ');
  Nhapdathuc(Dt2);
  Chia(Dt1,Dt2,Dt3,Dt4);Writeln;
  Writeln('Thuong cua hai da thuc la: ');
  Vietdathuc(Dt3);Writeln;
  Writeln('So du cua hai da thuc la: ');
  Vietdathuc(Dt4);
  Readln;
 END.

 7. Program P30507;
 {Tim nghiem huu ti cua da thuc bac 3 va phan tich no thanh 2 thua so}
 Uses Crt;
 Type  Dathuc = Record
```

```
            Deg : Byte;
            Koef: Array [0..100] of Real;
          End;
Var  Dt,Dt1,Dt2: Dathuc;
      X0: Real;
     Cn: Boolean;

Procedure Nhapdathuc(Var Dt: Dathuc);
Var i: Byte;
Begin
 FillChar(Dt,SizeOf(Dt),0);
 With Dt do
   begin
        Deg:=3;
        For i:=Deg downto 0 do
           begin
                 Write('Cho he so A[',i,'] cua da thuc: ');
                 Readln(Koef[i]);
            end;
   end;
End;

Function Tinhdathuc(Dt: Dathuc; X: Real): Real;
Var i: Byte; Y: Real;
Begin
 With Dt do
   begin
        Y:=Koef[Deg];
        For i:=Deg-1 downto 0 do Y:=X*Y+Koef[i];
   end;
 Tinhdathuc:=Y;
End;

Procedure Chuanhoa(Var Dt: Dathuc);
Var i: Byte;
Begin
While (Dt.Deg>0)and(Dt.Koef[Dt.Deg]=0) do Dt.Deg:=Dt.Deg-1;
End;

Procedure Hieu(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i: Byte;
Begin
```

```
 Dt3.Deg:=Dt2.Deg;
 If Dt1.Deg > Dt2.Deg then Dt3.Deg:=Dt1.Deg;
 For i:=0 to Dt3.Deg do Dt3.Koef[i]:=Dt1.Koef[i]-Dt2.Koef[i];
 Chuanhoa(Dt3);
End;

Procedure Tich(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 Dt3.Deg:=Dt1.Deg+Dt2.Deg;
 For i:=0 to Dt1.Deg do
   For j:=0 to Dt2.Deg do
Dt3.Koef[i+j]:=Dt3.Koef[i+j]+Dt1.Koef[i]*Dt2.Koef[j];
End;

Procedure Chia(Dt1,Dt2: Dathuc; Var Dt3,Dt4: Dathuc);
Var Dt,Dt5: Dathuc; i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 If Dt1.Deg<Dt2.Deg Then
    begin
          Dt4:=Dt1;
          Exit;
    end;
 Dt3.Deg:=Dt1.Deg-Dt2.Deg;
 Dt3.Koef[Dt1.Deg-Dt2.Deg]:=Dt1.Koef[Dt1.Deg]/Dt2.Koef[Dt2.Deg];
 Tich(Dt3,Dt2,Dt);
 Hieu(Dt1,Dt,Dt4);
 For i:=Dt1.Deg-1 downto Dt2.Deg do
   If Dt4.Deg >=Dt2.Deg then
    begin
          Dt3.Koef[i-Dt2.Deg]:=Dt4.Koef[Dt4.Deg]/Dt2.Koef[Dt2.Deg];
          Fillchar(Dt5,SizeOf(Dt5),0);
          Dt5.Deg:=Dt4.Deg-Dt2.Deg;
          Dt5.Koef[Dt5.Deg]:=Dt3.Koef[i-Dt2.Deg];
          Tich(Dt5,Dt2,Dt);
          Hieu(Dt4,Dt,Dt4);
    end;
End;

Procedure Tinh;
```

```
Var i,j: Byte;
Begin
 While Dt.Koef[0]=0 do
   begin
        For i:=0 to Dt.Deg-1 do  Dt2.Koef[i]:=Dt2.Koef[i+1];
        Dt2.Deg:=Dt.Deg-1;
        X0:=0;
        Cn:=True;
        Exit;
   end;
 For i:=1 to Round(Dt.Koef[Dt.Deg]) do
   If Round(Dt.Koef[Dt.Deg]) Mod i=0 then
     For j:=1 to Round(Dt.Koef[0]) do
        If Round(Dt.Koef[0]) Mod j=0 then
           begin
                X0:=j/i;
                If Tinhdathuc(Dt,X0)=0 then
                   begin
                         Dt1.Deg:=1;
                         Dt1.Koef[1]:=1;
                         Dt1.Koef[0]:=-X0;
                         Chia(Dt,Dt1,Dt2,Dt1);
                         Cn:=True;
                         Exit;
                   end;
                X0:=-X0;
                If Tinhdathuc(Dt,X0)=0 then
                   begin
                         Dt1.Deg:=1;
                         Dt1.Koef[1]:=1;
                         Dt1.Koef[0]:=-X0;
                         Chia(Dt,Dt1,Dt2,Dt1);
                         Cn:=True;
                         Exit;
                   end;
           end;
  End;

  Procedure Vietdathuc(Dt: Dathuc);
  Var i: Byte;
  Begin
   With Dt do
```

```
   begin
        Writeln('Bac cua da thuc la: ',Deg);
        For i:=Deg downto 0 do Writeln('He so A[',i,'] cua da thuc la:
',Koef[i]:10:2);
    end;
End;

BEGIN
 Clrscr;
 Cn:=False;
 Nhapdathuc(Dt);
 Tinh;
 If Cn then
  begin
       Writeln('Nghiem cua da thuc la: ',X0:10:2);
       Writeln('Thua so thu hai cua da thuc la: ');
       Vietdathuc(Dt2);
   end
 Else Writeln('Da thuc vo nghiem.');
 Readln;
END.
```

9. Program P30509; {Tim nguyen ham cua mot da thuc cho truoc}

```
Uses Crt;
Type Dathuc = Record
           Deg : Byte;
           Koef: Array [0..100] of Real;
          End;
Var  Dt1,Dt2: Dathuc;
     X: Real;

Procedure Nhapdathuc(Var Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
   begin
       Write('Cho bac cua da thuc: '); Readln(Deg);
       For i:= Deg downto 0 do
         begin
               Write('Cho he so A[',i,'] cua da thuc: ');
               Readln(Koef[i]);
          end;
```

```
  end;
End;

Procedure Vietdathuc(Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
   begin
        Writeln('Bac cua da thuc la: ',Deg);
        For i:= Deg downto 0 do  Writeln('He so A[',i,'] cua da thuc la:
',Koef[i]:10:2);
   end;
End;

Procedure Tinh;
Var i: Byte;
Begin
 Dt2.Deg:= Dt1.Deg+1;
 For i:= Dt1.Deg downto 0 do Dt2.Koef[i+1]:= Dt1.Koef[i]/(i+1);
End;

BEGIN
 Clrscr;
 Nhapdathuc(Dt1);
 Tinh;
 Vietdathuc(Dt2);
 Readln;
END.
```

11. Program P30511;

```
{Khao sat va ve do thi cua mot ham da thuc bac 4 - ham trung phuong}
Uses Crt,Graph;
Type  Dathuc = Record
              Deg : Byte;
              Koef: Array [0..100] of Real;
           End;
Var  X0,Y0,WX,WY,TX,TY: Integer;
     Dt: Dathuc;
     N: Byte;
     SX,FX,DX: Real;

Procedure MhDohoa;
```

```
Var ErrorCode,GraphDriver, GraphMode: Integer; Path: String;
Begin
 Clrscr;
 Path:= '';
 Repeat
   GraphDriver:=Detect;
   InitGraph(GraphDriver,GraphMode,Path);
   ErrorCode:=GraphResult;
   If ErrorCode<>grOk then
     begin
          Writeln('Loi do hoa:',GraphErrorMsg(ErrorCode));
          If Errorcode=GrFilenotFound then
             begin
                  Write('Go duong dan toi cac file BGI:');
                  Readln(Path);
             end
            Else
               begin
                    Readln;
                    Halt(1);
                end;
     end;
 Until ErrorCode =GrOK;
 SetColor(White);
 X0:=GetMaxX Div 2; Y0:=GetMaxY Div 2;
 Moveto(X0,0); LineTo(X0,GetMaxY);
 Moveto(0,Y0); LineTo(GetMaxX,Y0);
End;

Procedure Docdulieu;
Begin
 Write('Cho he so phong dai: '); Readln(N);
 Write('Cho diem bat dau ve: '); Readln(SX);
 Write('Cho diem ket thuc: ');   Readln(FX);
 Write('Cho gia so X: ');        Readln(DX);
End;

Procedure Chuanbi(X,Y: Real);
Begin
 Y:=-Y; X:=X*N; Y:=Y*N;
 X:=X+X0; Y:=Y+Y0;
 Tx:=Round(X); Ty:=Round(Y);
```

```
End;

Procedure Diden(X,Y: Real);
Begin
 Chuanbi(X,Y);
 Moveto(Tx,Ty);
End;

Procedure Veden(X,Y: Real);
Begin
 Chuanbi(X,Y);
 LineTo(Tx,Ty);
End;

Procedure Chiatruc;
Var i: integer;
Begin
 For i:=-(Y0 Div N) to (Y0 Div N) do
   begin
        MoveTo(X0-3,Y0+(i-1)*N);
        LineTo(X0+3,Y0+(i-1)*N);
   end;
 For i:=-(X0 Div N) to (X0 Div N) do
   begin
       MoveTo(X0+(i-1)*N,Y0-3);
       LineTo(X0+(i-1)*N,Y0+3);
   end;
 End;

 Procedure Nhapdathuc(Var Dt: Dathuc);
 Var i: Byte;
 Begin
  FillChar(Dt,SizeOf(Dt),0);
  Dt.Deg:=4;
  With Dt do
     begin
          For i:=2 downto 0 do
            begin
                 Write('Cho he so A[',i*2,'] cua da thuc: ');
                 Readln(Koef[i*2]);
              end;
     end;
```

```
End;
Procedure Daoham(Dt1: Dathuc; Var Dt2: Dathuc);
Var i: Byte;
Begin
 Dt2.Deg:=Dt1.Deg-1;
 For i:=Dt1.Deg downto 1 do Dt2.Koef[i-1]:=Dt1.Koef[i]*i;
End;
Procedure Vietdathuc(Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
   begin
        Writeln('Bac cua da thuc la: ',Deg);
        For i:=Deg downto 0 do  Writeln('He so A[',i,'] cua da thuc la:
',Koef[i]:10:2);
   end;
End;

Procedure Ptb2(A,B,C: Real; Var Delta,X1,X2: Real);
Begin
 Delta:=Sqr(B)-4*A*C;
 If Delta<0 then Exit;
 X1:=(-B+Sqrt(Delta))/(2*A);
 X2:=(-B-Sqrt(Delta))/(2*A);
End;

Function Tinhdathuc(Dt: Dathuc; X: Real): Real;
Var i: Byte; Y: Real;
Begin
 With Dt do
   begin
        Y:=Koef[Deg];
        For i:=Deg-1 downto 0 do Y:=X*Y+Koef[i];
    end;
 Tinhdathuc:=Y;
End;

Procedure Trinhbay;
Var X1,X2,Delta: Real; Dt1: Dathuc;
Begin
 Daoham(Dt,Dt1);
 Writeln('Dao ham cua da thuc la: ');
```

```
Vietdathuc(Dt1);
Ptb2(Dt.Koef[4],Dt.Koef[2],Dt.Koef[0],Delta,X1,X2);
If Delta >=0 then
  begin
       Writeln('Ham so co cac nghiem la: ');
       If X1>=0 then
       Write(Sqrt(X1),',',-Sqrt(X1));
       If X2>=0 then Writeln(Sqrt(X2),',',-Sqrt(X2));
   end;
PtB2(Dt1.Koef[3],Dt1.Koef[2],Dt1.Koef[1],Delta,X1,X2);
Writeln('Do thi co cac cuc tri tai:');
Writeln(0,',',Tinhdathuc(Dt,0));
If Delta>0 then
   begin
         Writeln(X1,',',Tinhdathuc(Dt,X1));
         Writeln(X2,',',Tinhdathuc(Dt,X2));
    end;
Write('Nhan Enter de tiep tuc..');
Readln;
End;

Procedure Ve;
Begin
 Diden(SX,Tinhdathuc(Dt,SX));
 While SX<FX Do
   begin
        SX:=SX+DX;
        Veden(SX,Tinhdathuc(Dt,SX));
   end;
End;

BEGIN
 Clrscr;
 Nhapdathuc(Dt);
 Trinhbay;
 Docdulieu;
 Mhdohoa;
 Chiatruc;
 Ve;
 Readln;
 CloseGraph;
END.
```

```
13. Program  P30513;
{Tim da thuc bac thap nhat la BSC cua hai da thuc da cho }
Uses Crt;
Type Dathuc = Record
            Deg : Byte;
            Koef: Array [0..100] of Real;
          End;
Var  Dt1,Dt2,Dt3: Dathuc;

Procedure Nhapdathuc(Var Dt: Dathuc);
Var i: Byte;
Begin
 FillChar(Dt,SizeOf(Dt),0);
 With Dt do
   begin
         Write('Cho bac cua da thuc: '); Readln(Deg);
         For i:=Deg downto 0 do
           begin
                 Write('Cho he so A[',i,'] cua da thuc: ');
                  Readln(Koef[i]);
            end;
   end;
End;

Procedure Vietdathuc(Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
    begin
          Writeln('Bac cua da thuc la: ',Deg);
          For i:=Deg downto 0 do Writeln('He so A[',i,'] cua da thuc la:
',Koef[i]:10:2);
    end;
End;

Procedure Chuanhoa(Var Dt: Dathuc);
Var i: Byte;
Begin
 While (Dt.Deg>0)and(Dt.Koef[Dt.Deg]=0) do Dt.Deg:=Dt.Deg-1;
End;
```

```
Procedure Hieu(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i: Byte;
Begin
 Dt3.Deg:=Dt2.Deg;
 If Dt1.Deg>Dt2.Deg then Dt3.Deg:=Dt1.Deg;
 For i:=0 to Dt3.Deg do Dt3.Koef[i]:=Dt1.Koef[i]-Dt2.Koef[i];
 Chuanhoa(Dt3);
End;

Procedure Tich(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 Dt3.Deg:=Dt1.Deg+Dt2.Deg;
 For i:=0 to Dt1.Deg do
   For j:=0 to Dt2.Deg do
Dt3.Koef[i+j]:=Dt3.Koef[i+j]+Dt1.Koef[i]*Dt2.Koef[j];
End;

Procedure Chia(Dt1,Dt2: Dathuc; Var Dt3,Dt4: Dathuc);
Var Dt,Dt5: Dathuc; i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 If Dt1.Deg<Dt2.Deg then
   begin
         Dt4:=Dt1;
         Exit;
    end;
 Dt3.Deg:=Dt1.Deg-Dt2.Deg;
 Dt3.Koef[Dt1.Deg-Dt2.Deg]:=Dt1.Koef[Dt1.Deg]/Dt2.Koef[Dt2.Deg];
 Tich(Dt3,Dt2,Dt);
 Hieu(Dt1,Dt,Dt4);
 For i:=Dt1.Deg-1 downto Dt2.Deg do
   If Dt4.Deg>=Dt2.Deg then
     begin
            Dt3.Koef[i-
Dt2.Deg]:=Dt4.Koef[Dt4.Deg]/Dt2.Koef[Dt2.Deg];
            Fillchar(Dt5,SizeOf(Dt5),0);
            Dt5.Deg:=Dt4.Deg-Dt2.Deg;
            Dt5.Koef[Dt5.Deg]:=Dt3.Koef[i-Dt2.Deg];
            Tich(Dt5,Dt2,Dt);
             Hieu(Dt4,Dt,Dt4);
```

```
    end;
End;

Procedure USC(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var Dt: Dathuc;
Begin
 If Dt1.Deg<Dt2.Deg then
   begin Dt:=Dt1; Dt1:=Dt2; Dt2:=Dt; end;
 Chia(Dt1,Dt2,Dt1,Dt);
 While (Dt.Deg>0)or(Dt.Koef[0] <> 0) do
   begin
        Dt:=Dt1; Dt1:=Dt2; Dt2:=Dt;
        Chia(Dt1,Dt2,Dt1,Dt);
   end;
 Dt3:=Dt2;
End;

Procedure BSC(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var Dt,Dtu: Dathuc;
Begin
 USC(DT1,DT2,Dtu);
 Chia(DT1,Dtu,Dt,Dt1);
 Tich(Dt,Dt2,Dt3);
End;

BEGIN
 Clrscr;
 Writeln('Nhap da thuc thu nhat: ');
 Nhapdathuc(Dt1);
 Writeln('Nhap da thuc thu hai: ');
 Nhapdathuc(Dt2);
 BSC(DT1,DT2,DT3);
 Writeln('Boi so chung cua hai da thuc la: ');
 Vietdathuc(Dt3);
 Readln;
END.
```

15. Program P30515; {Tinh dien tich}

```
Uses Crt;
Type Dathuc = Record
          Deg : Byte;
          Koef: Array [0..100] of Real;
```

```
        End;
Var  A,B,S: Real;
     Dt: Dathuc;
     N: Byte;

Procedure Nhapdathuc(Var Dt: Dathuc);
Var  i: Byte;
Begin
 With Dt do
   begin
         Write('Cho bac cua da thuc: '); Readln(Deg);
         For i:=Deg downto 0 do
            begin
                 Write('Cho he so A[',i,'] cua da thuc: ');
                 Readln(Koef[i]);
             end;
    end;
End;

Function Tinhdathuc(Dt: Dathuc; X: Real): Real;
Var i: Byte; Y: Real;
Begin
 With Dt do
  begin
        Y:=Koef[Deg];
        For i:=Deg-1 downto 0 do Y:=X*Y+Koef[i];
  end;
 Tinhdathuc:=Y;
End;

Procedure Tinh;
Var C,C1,C2: Real; i: Byte;
Begin
 S:=0; C1:=A; C:=(B-A)/N; C2:=C1+C;
 For i:=1 to N do
   begin
       S:=S+C*(Tinhdathuc(Dt,C1)+Tinhdathuc(Dt,C2))/2;
       C1:=C2; C2:= C1+C;
   end;
 End;

 BEGIN
```

```
 Clrscr;
 Nhapdathuc(Dt);
 Write('Nhap so A: ');
 Readln(A);
 Write('Nhap so B: ');
 Readln(B);
 Write('Nhap do chinh xac N: ');
 Readln(N);
 Tinh;
 Writeln('Dien tich la: ',S:10:2);
 Readln;
END.
```

17. Program P30517; {Tinh he so da thuc}

```
Uses Crt;
Type Dathuc = Record
            Deg : Byte;
            Koef: Array [0..100] of Real;
          End;
Var  Dt,Dt1,Dt2,Dt3: Dathuc;
     N: Byte;

Procedure Vietdathuc(Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
   begin
        Writeln('Bac cua da thuc la: ',Deg);
        For i:=Deg downto 0 do Writeln('He so A[',i,'] cua da thuc la:
',Koef[i]:10:2);
   end;
End;

Procedure Chuanhoa(Var Dt: Dathuc);
Var i: Byte;
Begin
 While Dt.Koef[Dt.Deg]=0 do Dt.Deg:=Dt.Deg-1;
End;

Procedure Hieu(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i: Byte;
Begin
```

```
Dt3.Deg:=Dt2.Deg;
If Dt1.Deg>Dt2.Deg then Dt3.Deg:=Dt1.Deg;
For i:=0 to Dt3.Deg do Dt3.Koef[i]:=Dt1.Koef[i]-Dt2.Koef[i];
Chuanhoa(Dt3);
End;

Procedure Tich(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 Dt3.Deg:=Dt1.Deg+Dt2.Deg;
 For i:=0 to Dt1.Deg do
   For j:=0 to Dt2.Deg do
Dt3.Koef[i+j]:=Dt3.Koef[i+j]+Dt1.Koef[i]*Dt2.Koef[j];
End;

Procedure Tinh;
Var i: Byte;
Begin
 Dt1.Deg:=0; Dt1.Koef[0]:=1;
 Dt2.Deg:=1; Dt2.Koef[1]:=1;
 Dt2.Koef[0]:=0;
 Dt.Deg:=1;
 Dt.Koef[0]:=0; Dt.Koef[1]:=2;
 If N=0 then  Dt3:=Dt1
    Else  Dt3:=Dt2;
 For i:=2 to N do
    begin
          Tich(Dt,Dt2,Dt3);
          Hieu(Dt3,Dt1,Dt3);
          Dt1:=Dt2; Dt2:=Dt3;
    end;
 End;

 BEGIN
  Clrscr;
  Write('Nhap so N: ');
  Readln(N);
  Tinh;
  Vietdathuc(Dt3);
  Readln;
 END.
```

```
19. Program P30519;  {Tinh he so da thuc Chebushev}
Uses Crt;
Type Dathuc = Record
            Deg : Byte;
            Koef: Array [0..100] of Real;
          End;
Var  Dt1,Dt2,Dt3: Dathuc;
     N: Byte;

Procedure Vietdathuc(Dt: Dathuc);
Var i: Byte;
Begin
 With Dt do
  begin
        Writeln('Bac cua da thuc la: ',Deg);
        For i:=Deg downto 0 do Writeln('He so A[',i,'] cua da thuc la:
',Koef[i]:10:2);
   end;
End;

Procedure Chuanhoa(Var Dt: Dathuc);
Var i: Byte;
Begin
 While Dt.Koef[Dt.Deg] =0 do Dt.Deg:=Dt.Deg-1;
End;

Procedure Hieu(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i: Byte;
Begin
 Dt3.Deg:=Dt2.Deg;
 If Dt1.Deg>Dt2.Deg then Dt3.Deg:=Dt1.Deg;
 For i:=0 to Dt3.Deg do Dt3.Koef[i]:=Dt1.Koef[i]-Dt2.Koef[i];
 Chuanhoa(Dt3);
End;

Procedure Tich(Dt1,Dt2: Dathuc; Var Dt3: Dathuc);
Var i,j: Byte;
Begin
 FillChar(Dt3,SizeOf(Dt3),0);
 Dt3.Deg:=Dt1.Deg+Dt2.Deg;
 For i:=0 to Dt1.Deg do
```

```
  For j:=0 to Dt2.Deg do
Dt3.Koef[i+j]:=Dt3.Koef[i+j]+Dt1.Koef[i]*Dt2.Koef[j];
End;
Procedure Tinh;
Var i: Byte; Dt: Dathuc;
Begin
 Dt1.Deg:=0; Dt1.Koef[0]:=1;
 Dt2.Deg:=1; Dt2.Koef[1]:=1;
 Dt2.Koef[0]:=0;
 Dt.Deg:=1; Dt.Koef[1]:=2;
 Dt.Koef[0]:=0;
 If N=0 then Dt3:=Dt1;
 If N=1 then Dt3:=Dt2;
 For i:=2 to N do
   begin
         Tich(Dt,Dt2,Dt);
         Hieu(Dt,Dt1,Dt3);
         Dt1:=Dt2; Dt2:=Dt3;
    end;
End;
BEGIN
 Clrscr;
 Write('Cho so N: ');
 Readln(N);
 Tinh;
 Vietdathuc(Dt3);
 Readln;
END.
```

Bài 6. Biến đổi ma trận

```
1. Program P30601; {Hop cua hai ma tran nhi phan}
Uses Crt;
Const   fi='Mt.inp';
        fo1='hop.out';
        max=100;
Var A,B,H: Array [1..max,1..max] of byte;
    F : Text;
    m,n : Byte;

Procedure Nhap;
Var i,j: byte;
Begin
  Assign(F,fi); Reset(f);
  Readln(f,m,n);
  For i:=1 to m do
    begin
        For j:=1 to n do Read(F,A[i,j]);
        Readln(F);
    end;
  For i:=1 to m do
   begin
       For j:=1 to n do Read(F,B[i,j]);
       Readln(F);
   end;
  Close(f);
End;

Procedure Hop;
Var i,j: byte;
Begin
For i:=1 to m do
  For j:=1 to n do H[i,j]:=a[i,j] or B[i,j];
End;

Procedure Xuly;
Begin
  Hop;
End;
```

```
Procedure Ketqua;
Var i,j: byte;
Begin
Assign(F,fo1); Rewrite(f);
For i:=1 to m do
 begin
     For j:=1 to n do Write(f,h[i,j],' ');
     Writeln(f);
  end;
Readln;
Close(f);
End;

BEGIN
 Clrscr;
 Nhap;
 Xuly;
 Ketqua;
END.
```

3. Program P30603; {Tich logic cua hai ma tran}

```
Uses Crt;
Const   fi='Mt3.inp';
        fo='Mt3.out';
        max=100;
Var A,B,C: Array [1..max,1..max] of byte;
     f : Text;
    m,n,k: Byte;

Procedure Nhap;
Var i,j: byte;
Begin
Assign(f,fi); Reset(f);
Readln(f,m,k,n);
For i:=1 to m do
  begin
       For j:=1 to k do Read(f,a[i,j]);
       Readln(f);
   end;
For i:=1 to k do
  begin
```

```
        For j:=1 to n do Read(f,B[i,j]);
        Readln(f);
    end;
Close(f);
End;

Procedure Xuly;
Var i,j,t,p: byte;
Begin
 For i:=1 to m do
  For j:=1 to n do
   begin
        p:=a[i,1] and b[1,j];
        For t:=2 to k do p:=p or (a[i,t] and b[t,j]);
        c[i,j]:=p;
    end;
End;

Procedure Ketqua;
Var i,j: byte;
Begin
 Assign(f,fo); Rewrite(f);
 For i:=1 to m do
  begin
      For j:=1 to n do Write(f,c[i,j],' ');
      Writeln(f);
   end;
 Close(f);
End;

BEGIN
 Clrscr;
 Nhap;
 Xuly;
 Ketqua;
END.
```

5. Program P30605; {Cac phep toan voi ma tran}

```
Uses Crt;
Const   max=100;
        fi1='chuyenvi.inp';
        fi2='tong.inp';
        fi3='tich.inp';
        fo1='chuyenvi.out';
        fo2='tong.out';
        fo3='tich.out';
Var   A,B,C: Array [1..max,1..max] of integer;
      m,n,k: Byte;
      F : Text;

Procedure Chuyenvi;
Var i,j: byte;
Begin
 Assign(f,fi1); Reset(f);
 Readln(f,m,n);
 For i:=1 to m do
  begin
       For j:=1 to n do Read(f,a[i,j]);
       Readln(f);
  end;
 Close(f);
 Assign(f,fo1); Rewrite(f);
 For i:=1 to n do
  begin
       For j:=1 to m do Write(f,a[j,i],' ');
       Writeln(f);
  end;
 Close(f);
End;

Procedure Tong;
Var i,j: byte;
Begin
 Assign(f,fi2); Reset(f);
 Readln(f,m,n);
```

```
 For i:=1 to m do
  begin
       For j:=1 to n do Read(f,a[i,j]);
       Readln(f);
   end;
 For i:=1 to m do
    begin
        For j:=1 to n do Read(f,b[i,j]);
        Readln(f);
    end;
 Close(f);
 Assign(f,fo2); Rewrite(f);
 For i:=1 to m do
   begin
        For j:=1 to n do Write(f,a[i,j]+b[i,j],' ');
        Writeln(f);
    end;
 Close(f);
End;

Procedure Tich;
Var i,j,t: byte; gt: longint;
Begin
 Assign(f,fi3); Reset(f);
 Readln(f,m,k,n);
 For i:=1 to m do
  begin
       For j:=1 to k do Read(f,a[i,j]);
       Readln(f);
   end;
 For i:=1 to k do
  begin
       For j:=1 to n do Read(f,b[i,j]);
       Readln(f);
```

```
  end;
 Close(f);
 Assign(f,fo3); Rewrite(f);
 For i:=1 to m do
  For j:=1 to n do
    begin
        gt:=0;
        For t:=1 to k do gt:=gt+a[i,t]*b[t,j];
        c[i,j]:=gt;
     end;
 For i:=1 to m do
   begin
        For j:=1 to n do Write(f,c[i,j],' ');
        Writeln(f);
    end;
  Close(f);
End;

Procedure Xuly;
Begin
 Chuyenvi;
 Tong;
 Tich;
End;

BEGIN
 Clrscr;
 Xuly;
END.
```

7. Program P30607; {Tinh dinh thuc bang hoan vi}

```
Uses Crt;
Const   fi = 'P30607.inp';
Var  a : array [1..20,1..20] of real;
       b,dx: array [1..20] of byte;
       det : real;
       n : byte;

Procedure Input;
Var f: text; i,j: byte;
Begin
```

```
 Assign(f,fi); Reset(f);
 Readln(f,n);
 For i:=1 to n do
  begin
       For j:=1 to n do Read(f,a[i,j]);
       Readln(f);
   end;
 Close(f);
End;

Function Dau: integer;
Var d,i,j,count: integer;
Begin
 count:=0;
 For i:=1 to n-1 do
   For j:=i+1 to n do If b[i]>b[j] then inc(count);
 If (count mod 2)=0 then dau:=1 Else dau:=-1;
End;

Procedure Tinh(dem: byte);
Var i: byte; t: real;
Begin
 If dem>n then
  begin
        t:=1;
        For i:=1 to n do
          begin
               Write(b[i],' ');
               t:=t*a[i,b[i]];
           end;
        det:=det+dau*t;
        Writeln;
        Exit;
  end;
 For i:=1 to n do
   If dx[i]=0 then
      begin
            dx[i]:=1;
            b[dem]:=i;
            tinh(dem+1);
            dx[i]:=0;
```

```
      end;
End;

BEGIN
  Clrscr;
  Input;
  Fillchar(dx,Sizeof(dx),0);
  det:=0;
  Tinh(1);
  Write(det);
  Readln;
END.
```

9.
```
{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+}
{$M 16384,0,655360}
Program P30609; {Ma tran nghich dao}
Uses Crt;
Const   max=10;
        fi = 'hept.Inp';
        fo = 'hept.Out';
Type  mang = array [1..max,1..max] of real;
Var A,B,C: mang;
    hs,D: Array [1..max] of real;
    n : byte;
    f : Text;
    gt : real;

Procedure Nhap;
Var i,j: byte;
Begin
  Assign(f,fi); Reset(f);
  Readln(f,N);
  For i:=1 to n do
    begin
        For j:=1 to n do Read(f,a[i,j]);
        Readln(f);
    end;
  For i:=1 to n do Read(f,hs[i]);
  Close(f);
End;
```

```
Procedure Bo(kt: byte; Var x: mang; z,y: byte);
Var i,j: byte;
Begin
 For i:=1 to kt do
  For j:=y to kt-1 do X[i,j]:=X[i,j+1];
 For i:=z to kt-1 do
  For j:=1 to kt-1 do X[i,j]:=X[i+1,j];
End;

Function Det(x: byte; bt: mang; p,q: byte): real;
Var i: byte; gt1,gt2: real;
Begin
 If (p<>0) then bo(x+1,bt,p,q);
 If x=1 then
   begin
        det:=bt[1,1];
        exit;
    end;
 For i:=1 to x do
  begin
       gt1:=0; gt2:=0;
       If i mod 2=1 then gt1:=Bt[1,i]*det(x-1,Bt,1,i)
         Else gt2:=-Bt[1,i]*det(x-1,Bt,1,i);
       gt:=gt+gt1+gt2;
   end;
 det:=gt;
End;

Procedure Lam;
Var i,j: byte; g,t: real;
Begin
 Assign(f,fo); Rewrite(f);
 C:=A; gt:=0;
 g:=det(n,C,0,0);
 If g=0 then
  begin
        Writeln(f,'Khong kha nghich');
        Close(f);
        Halt;
   end;
 For i:=1 to n do
   For j:=1 to n do
```

```
    begin
          C:=a;
          gt:=0;
          t:=det(n-1,C,j,i);
          If i+j mod 2=0 then B[i,j]:=t/g
            Else B[i,j]:=-1*t/g;
     end;
 Close(f);
End;

Procedure Nhan;
Var i,j,t: byte;
Begin
 Assign(f,fo); Reset(f);
 While not Eof(f) do
   begin
       For i:=1 to n do
         begin
               gt:=0;
               For t:=1 to n do gt:=gt+B[i,t]*hs[t];
               D[i]:=gt;
         end;
       For i:=1 to n do
         begin
               For j:=1 to 1 do Write(f,D[i],' ');
               Writeln(f);
          end;
   end;
End;

Procedure Xuly;
Begin
 Lam;
 Nhan;
End;

Procedure Ketqua;
Var i,j: byte;
Begin
 Assign(f,fo); Reset(f);
 While not Eof(f) do
```

```
  For i:=1 to n do
    begin
        For j:=1 to n do Write(f,b[i,j]:10:5,' ');
        Writeln(f);
    end;
 Close(f);
End;
BEGIN
 Clrscr;
 Nhap;
 Xuly;
 Ketqua;
END.
```

13. Program P30613; {Tinh A^2}

```
Uses Crt;
Const  max=50;
       fi='nhanMT.Inp';
       fo='NhanMt.Out';
Var  A,B,C: Array [1..Max,1..max] of Longint;
     N,Mu : Byte;
      f : Text;

Procedure Nhap;
Var i,j: byte;
Begin
 Assign(f,fi); Reset(f);
 Readln(f,N,Mu);
 For i:=1 to N do
  begin
      For j:=1 to N do Read(f,A[i,j]);
      Readln(f);
  end;
 Close(f);
End;

Procedure Xuly;
Var bac,i,j,k: byte; gt: longint;
Begin
 bac:=1; C:=a;
```

```
 While 2*bac<mu do
  begin
      B:=A;
      For i:=1 to n do
       For j:=1 to n do
        begin
             Gt:=0;
             For k:=1 to n do gt:=gt+B[i,k]*b[k,j];
             A[i,j]:=gt;
         end;
      bac:=2*bac;
  end;
 While bac<mu do
  begin
      B:=A;
      For i:=1 to n do
       For j:=1 to n do
         begin
               Gt:=0;
               For k:=1 to n do gt:=gt+B[i,k]*C[k,j];
               A[i,j]:=gt;
         end;
      bac:=bac+1;
  end;
End;

Procedure Ketqua;
Var i,j: byte;
Begin
 Assign(f,fo); Rewrite(f);
 For i:=1 to n do
   begin
        For j:=1 to n do Write(f,A[i,j],' ');
        Writeln(f);
   end;
 Close(f);
End;

BEGIN
 Clrscr;
 Nhap;
```

```
  Xuly;
  Ketqua;
END.
{
Ví dụ  Input:
3 7
1 2 3
2 3 1
3 1 2
Ví dụ Output:
93285 93312 93339
93312 93339 93285
93339 93285 93312 }
```

15. Xem bài **13** (P30613).

17.
```
{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+,Y+}
{$M 16384,0,655360}
Program P30617; {Kiem tra ma tran co kha nghich hay khong}
Uses Crt;
Const   max=100;
         Fi = 'bdmt.Inp';
         Fo = 'Checkkn.Out';
Var  A : Array [1..max,1..max+1] of real;
     X : Array [1..max] of real;
     n : byte;
     f,f1: Text;

Procedure Nhap;
Var i,j: byte;
Begin
 Assign(f,fi); Reset(f);
 Readln(f,N);
 For i:=1 to n do
  begin
       For j:=1 to n do Read(f,a[i,j]);
       Readln(f);
  end;
 Close(f);
End;
```

```
Procedure Ketthuc;
Begin
 Assign(f1,fo); Rewrite(f1);
 Writeln(f1,'Khong kha nghich');
 Close(f1);
End;

Function Tim(x: byte): byte;
Var i: byte;
Begin
 tim:= 0;
 For i:=x+1 to n do
   If a[i,x]<>0 then begin Tim:=i; exit; end;
End;

Procedure Bdcs1h(j,y: byte);
Var i: byte; tg: real;
Begin
 For i:=1 to n+1 do
   begin
       tg:=a[j,i];
       a[j,i]:=a[y,i];
       a[y,i]:=tg;
   end;
End;

Procedure Bdcs3h(y,j: byte; nho: real);
Var i: byte;
Begin
 For i:=1 to N+1 do A[y,i]:=a[y,i]-nho*a[j,i];
End;

Procedure Bdcs2h(x: byte; nho: real);
Var i: byte;
Begin
 For i:=1 to n do A[x,i]:=a[x,i]*nho;
End;

Procedure Khu_lan1;
Var i,j: byte; nho: real;
Begin
 For i:=1 to n do
```

```
  begin
      If a[i,i]=0 then
        begin
            j:=tim(i);
            If j<>0 then Bdcs1h(i,j)
              Else begin Ketthuc; Halt; end;
        end;
      Bdcs2H(i,1/a[i,i]);
      For j:=i+1 to n do Bdcs3H(j,i,a[j,i]);
  end;
End;

Procedure Khu_lan2;
Var i,j: byte;
Begin
 For i:=1 to N-1 do
   For j:=i+1 to n do bdcs3h(i,j,a[i,j]);
End;

Procedure Xuly;
Begin
 Khu_lan1;
 Khu_lan2;
 Assign(f1,fo); Rewrite(f1);
 Writeln(f1,'kha nghich');
 Close(f1);
End;

BEGIN
 Clrscr;
 Nhap;
 Xuly;
END.
```

19. {$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+,Y+}

```
{$M 16384,0,655360}
Program P30619; {Tim Ma tran nguoc}
Uses Crt;
Const  max=20;
       Fi='bdmt.Inp';
Var  A,B: Array [1..max,1..max+1] of real;
     X : Array [1..max] of real;
```

```
    n : byte;
    f : Text;

Procedure Nhap;
Var i,j: byte;
Begin
  Assign(f,fi); Reset(f);
  Readln(f,n);
  For i:=1 to n do
    begin
        For j:=1 to n do Read(f,a[i,j]);
        Readln(f);
    end;
  Close(f);
  For i:=1 to n do
   For j:=1 to n do If i=j then B[i,j]:=1 Else B[i,j]:=0;
End;

Procedure Xuat;
Var i,j: byte;
Begin
  For i:=1 to n do
    begin
        For j:=1 to n do Write(a[i,j]:8:2); Write(' ');
        For j:=1 to n do Write(b[i,j]:8:2); Writeln;
    end;
  Readln;
End;

Procedure Ketthuc;
Begin
  Writeln('Khong co ma tran nghich dao!');
End;

Function Tim(x: byte): byte;
Var i: byte;
Begin
  Tim:= 0;
   For i:=x+1 to n do
     If a[i,x]<>0 then begin Tim:= i; exit; end;
End;
```

```
Procedure Bdcs1h(j,y: byte);
Var i: byte; tg: real;
Begin
 For i:=1 to n+1 do
  begin
       tg:=a[j,i];
       a[j,i]:=a[y,i];
       a[y,i]:=tg;
  end;
 For i:=1 to n+1 do
  begin
       tg:=b[j,i];
       b[j,i]:=b[y,i];
       b[y,i]:=tg;
  end;
 Writeln('BDCS1 H ',j,' ',y);
Xuat;
End;

Procedure Bdcs3h(y,j: byte; nho: real);
Var i: byte;
Begin
 For i:=1 to N+1 do A[y,i]:=a[y,i]-nho*a[j,i];
 For i:=1 to N+1 do b[y,i]:=b[y,i]-nho*b[j,i];
 Writeln('BDCS3 H ',y,' - ', nho:10:2 ,'*',j);
 Xuat
End;

Procedure Bdcs2h(x: byte; nho: real);
Var i: byte;
Begin
 For i:=1 to n do A[x,i]:=a[x,i]*nho;
 For i:=1 to n do b[x,i]:=b[x,i]*nho;
 Writeln('BDCS2 H ',X,'*',nho:10:2);
 Xuat;
End;

Procedure Khu_lan1;
Var i,j: byte; nho: real;
Begin
 For i:=1 to n do
```

```
  begin
      If a[i,i]=0 then
        begin
             j:=tim(i);
             If j<>0 then Bdcs1h(i,j)
              Else begin Ketthuc; Halt; end;
        end;
      Bdcs2H(i,1/a[i,i]);
      For j:=i+1 to n do Bdcs3H(j,i,a[j,i]);
   end;
End;

Procedure Khu_lan2;
Var i,j: byte;
Begin
  For i:=1 to N-1 do
    For j:=i+1 to n do bdcs3h(i,j,a[i,j]);
End;

Procedure Xuly;
Begin
  Khu_lan1;
  Khu_lan2;
End;

BEGIN
  Clrscr;
  Nhap;
  Xuat;
  Xuly;
END.
```

Bài 7. Số ngẫu nhiên. Tính xấp xỉ giá trị số

1. Program P30701; {Giai xap xi phuong trinh}

```
Uses Crt;
Var a,b,x,ex,fa,fb,fx: real;

Function f(x: real): real;
Begin
  f:=x*x*x+x*x+x+1;
End;

BEGIN
Clrscr;
ex:=0.0001; a:=-5; b:=5;
 Repeat
   x:=(a+b)/2;
   fa:=f(a); fb:=f(b); fx:=f(x);
   If fa*fx>0 then a:=x
     Else If fb*fx>0 then b:=x;
 Until abs(fx)<ex;
 Writeln(x:0:4);
 Readln;
END.
```

3. Program P30703; {Giai xap xi phuong trinh}

```
Uses Crt;
Var a,b,x,ex,fa,fb,fx: real;

Function f(x: real): real;
Begin
 f:=x*x-sin(5*x);
End;

BEGIN
 Clrscr;
 ex:=0.01; a:=0.5; b:=0.6;
 Repeat
   x:=(a+b)/2;
   fa:=f(a); fb:=f(b); fx:=f(x);
```

```
  If fa*fx>0 then a:=x
    Else If fb*fx>0 then b:=x;
 Until abs(fx)<ex;
 Writeln(x:0:2);
 Readln;
END.
```

5. Program P30705; {Giai xap xi phuong trinh}

```
Uses Crt;
Var x0,x: real;

Function f(x: real): real;
Begin
 f:=x*x*x-2*x*x+x-3;
End;

Function df(x: real): real;
Begin
 df:=3*x*x-4*x+1;
End;

BEGIN
 Clrscr;
 x0:=2.2;
 Repeat
   x0:=x;
   x:=x-f(x)/df(x);
 Until Abs(x-x0)<0.0001;
 Write(x:0:4);
 Readln;
END.
```

11. Program P30711; {Tinh tich phan}

```
Uses Crt;
Var a,b: real;

Function f(x: real): real;
Begin
 f:=3;
End;

Function Inter1(a,b: real; n: longint): real;
```

```
Var h,t: real; i: longint;
Begin
 h:=(b-a)/n;
 t:=0;
 For i:=0 to n do t:=t+f(a+i*h+h/2);
 inter1:=t*h;
End;

Function Inter(a,b: real): real;
Var n0,i: longint; e,t,t1: real;
Begin
 e:=0.001;
 n0:=10;
 t:=inter1(a,b,n0);
 i:=1;
 Repeat
  t1:=t;
  t:=inter1(a,b,2*i*n0);
  Inc(i);
 Until abs(t-t1)<e;
 inter:=t;
End;

BEGIN
 Clrscr;
 Write(' A = ');Readln(a);
 Write(' B = ');Readln(b);
 Write(inter(a,b):0:10);
 Readln;
END.
```

19. Program P30719; { Sinh day so thap phan ngau nhien}

```
Uses Crt;
Var k,i: word;
     s : String;

BEGIN
 Clrscr;
 Write(' Nhap K: '); Readln(k);
 s:='';
 Randomize;
```

```
 For i:=1 to k do s:=s+chr(random(9)+49);
 Write(s);
 Readln;
END.
```

21. Program P30721;

```
{Sinh day so ngau nhien phan bo deu cua (0,1,2,3,4,5,6,7,8,9)}
Uses Crt;
Const   a: array [1..10] of integer = (0,1,2,3,4,5,6,7,8,9);
Var v,i: integer;
     x: real;
     p,ii: array [0..10] of real;

BEGIN
 Clrscr;
 ii[0]:=0;
 For i:=1 to 10 do
   begin
        p[i]:=1/10;
        ii[i]:=ii[i-1]+1/10;
   end;
 Repeat
   x:=random;
   For i:=1 to 10 do If (x>=ii[i-1]) and (x<ii[i]) then v:=a[i];
   Write(v);
 Until Keypressed;
END.
```

Bài 8. Xử lí văn bản

1. Program P30801;
{Liet ke chu cai co trong file dung mot lan theo thu tu}
Uses Crt;
Const MN = 255;
fn = 'P30802.inp';
gn = 'P30802.out';
Type mang1 = array [0..255] of integer;
Var a : mang1;
f,g: text;

Procedure Doc;
Var c: char;
Begin
Assign(f,fn); Reset(f);
FillChar(a,SizeOf(a),0);
While not Eof(f) do
begin
While not Eoln(f) do
begin
Read(f,c); c:=UpCase(c);
Inc(a[ord(c)]);
end;
Readln(f);
end;
Close(f);
End;

Procedure Viet;
Var i: integer;
Begin
Assign(g,gn); Rewrite(g);
For i:=0 to 255 do
If chr(i) in ['a'..'z','A'..'Z'] then
If a[i]=1 then Writeln(g,chr(i));
Close(g);
End;

BEGIN
Clrscr;

```
 Doc;
 Viet;
END.
```

3. Program P30803;

```
Uses  Crt;
Const   MN = 255;
         fn = 'P30803.inp';
         gn = 'P30803.out';
Type    mang1 = array [0..255] of integer;
Var   a  : mang1;
      f,g: text;

Procedure Doc;
Var c: char;
Begin
 Assign(f,fn); Reset(f);
 FillChar(a,SizeOf(a),0);
 While not Eof(f) do
   begin
        While not Eoln(f) do
           begin
                Read(f,c); c:=UpCase(c);
                Inc(a[ord(c)]);
           end;
        Readln(f);
   end;
 Close(f);
End;

Procedure Viet;
Var i,tong: integer;
Begin
 Assign(g,gn); Rewrite(g);
 Tong:=0;
 For i:=0 to 255 do
  If chr(i) in ['a'..'z','A'..'Z'] then Tong:=Tong+a[i];
 For i:=0 to 255 do
  If chr(i) in ['a'..'z','A'..'Z'] then
    If a[i]<>0 then Writeln(g,chr(i),': ',a[i]/Tong*100:4:4,' %');
 Close(g);
End;
```

```
BEGIN
 Clrscr;
 Doc;
 Viet;
END.
```

5. Program P30805; {Tinh ti le cac nguyen am}

```
Uses  Crt;
Const   MN = 255;
         fn = 'P30805.inp';
        gn = 'P30805.out';
Type  mang1 = array[0..255] of integer;
Var   a  : mang1;
      f,g: text;

Procedure Doc;
Var c: char;
Begin
 Assign(f,fn); Reset(f);
 FillChar(a,SizeOf(a),0);
 While not Eof(f) do
    begin
       While not Eoln(f) do
          begin
                Read(f,c); c:=UpCase(c);
                Inc(a[ord(c)]);
           end;
       Readln(f);
    end;
 Close(f);
End;

Procedure Viet;
Var i,tong: integer;
Begin
 Assign(g,gn); Rewrite(g);
 tong:=0;
 For i:=0 to 255 do
   If chr(i) in ['a'..'z','A'..'Z'] then tong:=tong+a[i];
 For i:=0 to 255 do
   If chr(i) in ['A','O','E','U','I','Y'] then
```

```
   If a[i]<>0 then Writeln(g,chr(i),': ',a[i]/tong*100:4:4,' %');
 Close(g);
End;

BEGIN
 Clrscr;
 Doc;
 Viet;
END.
```

7. Program P30807;

```
{Kiem tra sap xep theo chieu giam cua bac don thuc}
Uses  Crt;
Const   MN = 100;
         fn ='P30807.inp';
         gn ='P30807.out';
Type  mangR = array [0..100] of real;
Var  a  : mangr;
      sa : real;
      sb : integer;
      dau: char;
      s : string;
      si : integer;
      f,g: text;

Procedure Nhap;
Begin
 Assign(f,fn); Reset(f);
 Readln(f,s);
 Close(f);
End;

Procedure Sapxep;
Begin

End;

Procedure Xoacach;
Var i: integer;
Begin
 i:=1;
 While i<=length(s) do
```

```
  If s[i]=' ' then Delete(s,i,1) Else Inc(i);
 If (s[1]<>'-')and(s[1]<>'+') then s:='+'+s ;
End;

Procedure Loi(x: integer);
Begin
 Case x of
   1: Write('thieu dau');
   2: Write('thieu a');
   3: Write('thieu b');
 End;
 Readln;
 Halt;
End;

Procedure Docdau;
Begin
 If not (s[si] in ['+','-']) then Loi(1)
   Else begin dau:=s[si]; Inc(si); end;
End;

Procedure Doca;
Var d: array [1..2] of longint; cho: integer; t: real;
Begin
 FillChar(d,SizeOf(d),0);
 cho:=1;
 If not (s[si] in ['0'..'9']) then Loi(2);
 While (si<=length(s))and(s[si] in ['0'..'9','.']) do
   begin
      Case s[si] of
           '0'..'9': d[cho]:=d[cho]*10+ord(s[si])-ord('0');
            '.'     : cho:=2;
          End;
       Inc(si);
   end;
 t:=d[2];
 While t>1 do t:=t/10;
 sa:=d[1]+t;
End;

Procedure Docb;
Begin
```

```
sb:=0;
If s[si]<>'x' then sb:=0
 Else
    begin
          Inc(si);
          If s[si] <>'^' then begin sb:=1; exit; end;
          Inc(si);
          If not(s[si] in ['0'..'9']) then Loi(3);
          While (si<=length(s))and(s[si] in ['0'..'9']) do
             begin
                   sb:=sb*10+ord(s[si])-ord('0');
                   Inc(si);
              end;
    end;
End;

Procedure Nhapdab;
Begin
 If dau='-' then sa:=-sa;
 a[sb]:=a[sb]+sa;
End;

Procedure Phantich;
Begin
 si:=1;
 While si<=length(s) do
   begin
        Docdau;
        Doca;
        Docb;
        Nhapdab;
   end;
End;

Procedure Xuly;
Begin
 FillChar(a,SizeOf(a),0);
 Xoacach;
 Phantich;
End;

Procedure Viet;
```

```
Var i: integer;
Begin
 Assign(g,gn); Rewrite(g);
 For i:=MN downto 0 do
   If abs(a[i])>0.0001 then
     If a[i]>0 then Write(g,'+ ',a[i]:4:2,'x^',i)
       Else Write(g,a[i]:4:2,'x^',i);
 Close(g);
End;

BEGIN
 Clrscr;
 Nhap;
 Xuly;
 Viet;
END.
```

9.

```
{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+}
{$M 16384,0,655360}
Program P30809;
{Kiem tra xem xau nhap vao co phai la bieu thuc so hoc khong}
Uses  Crt;
Const MN = 1000;
        fn = 'P30809.INP';
        gn = 'P30809.OUT';
        khac = #255;
Type  mang1 = array [1..MN] of integer;
Var   p,gt,st : mang1;
      s : string;
      pi,so,si,sst: integer;
      sym : char;
      f,g : text;

Procedure Motep;
Begin
 Assign(f,fn); Reset(f);
 Assign(g,gn); Rewrite(g);
End;

Procedure Dongtep;
Begin
 Close(f); Close(g);
```

```
End;

Procedure Init;
Begin
 pi:=0;
 si:=1;
 s:=s+khac;
 sst:=0;
End;

Procedure Loi;
Begin
End;

Procedure Docso;
Begin
 so:=0;
 While s[si] in['0'..'9'] do
   begin
         so:=so*10+ord(s[si])-ord('0');
         Inc(si);
   end;
End;

Procedure Cach;
Begin
 While s[si]=' 'do Inc(si);
End;

Procedure Getsym;
Begin
 Cach;
 Case s[si] of
    '+','-','*','/','(',')' : begin sym:=s[si]; Inc(si); end;
    '0'..'9'               : begin sym:='S';  Docso;  end;
   Else sym:=khac;
 End;
End;

Procedure Nhapdau(d: char);
Begin
 Inc(pi); p[pi]:=2;
```

```
 Case d of
  '+': gt[pi]:=1;
  '-': gt[pi]:=2;
  '*': gt[pi]:=3;
  '/': gt[pi]:=4;
  '&': gt[pi]:=5;
 End;
End;

Procedure Nhapso;
Begin
 Inc(pi); p[pi]:=1; gt[pi]:=so;
End;

Procedure Bieuthuc;
Var daubt: char;

 Procedure Hangtu;
 Var dauht: char;

  Procedure Nhantu;
  Begin      {1.1}
   Case sym of
    'S': Nhapso;
    '(': begin {2.1}
           getsym;
           Bieuthuc;
           If sym<>')' then Loi;
         end;  {2.1}
   End;  {case}
   getsym;
  End;{1.1}

Begin {1.2}
 Clrscr;
 Nhantu;
 While sym in ['*','/'] do
  begin   {2.2}
      dauht:=sym;
      getsym;
      nhantu;
      nhapdau(dauht);
```

```
  end;   {2.2}
End;  {1.2}

Begin   {1.3}
 If sym in ['+','-'] then
  begin     {2.3}
      If sym='-' then daubt:=sym;
      getsym;
  end;      {2.3}
 hangtu;
 If daubt='-' then Nhapdau('&');
 While sym in ['+','-'] do
   begin      {3.3}
        daubt:=sym;
        getsym;
        Hangtu;
        Nhapdau(daubt);
   end;        {3.3}
 End;{1.3}

 Procedure Xem;
 Var i: integer;
 Begin
  For i:=1 to pi do
   If p[i]=1 then Writeln(gt[i])
    Else  Case gt[i] of
              1: Writeln('+');
              2: Writeln('-');
              3: Writeln('*');
              4: Writeln('/');
              5: Writeln('-');
           End;
 End;

 Procedure Thuchien(x: integer);
 Begin
  Case x of
     1: begin st[sst-1]:=st[sst-1]+st[sst]; Dec(sst); end;
     2: begin st[sst-1]:=st[sst-1]-st[sst]; Dec(sst); end;
     3: begin st[sst-1]:=st[sst-1]*st[sst]; Dec(sst); end;
     4: begin st[sst-1]:=st[sst-1] div st[sst]; Dec(sst); end;
     5: st[sst]:=-st[sst];
```

```
  End;
End;

Procedure Tinh;
Var i: integer;
Begin
 For i:=1 to pi do
  If p[i]=1 then begin Inc(sst); st[sst]:=gt[i] end
    Else Thuchien(gt[i]);
End;

Procedure Viet;
Begin
 Writeln(g,st[1]);
End;

BEGIN
 Clrscr;
 Motep;
 While not Eof(f) do
   begin
     Clrscr;
     Readln(f,s);
     Write(g,s,' = ');
     Init;
     Getsym;
     Bieuthuc;
     Tinh;
     Viet;
  end;
 Dongtep;
END.
```

11. Program P30811; {Kiem tra bieu thuc dai so}

```
Uses Crt;
Var  s: String;

Function test(s: String): boolean;
Var i,dem,dau: integer;
Begin
 test:=false;
 dem:=0;
```

```
dau:=1;
i:=1;
While (i<=length(s)) do
 begin
    Case s[i] of
          '('     : begin If dau=0 then exit; inc(dem); end;
          ')'     : begin
                          If dau=1 then exit;
                          Dec(dem);
                           If dem<0 then exit;
                     end;
    '+','-','*','/'  : If dau=1 then exit
                     Else dau:=1;
   'A'..'Z','a'..'z': begin
                       If dau=0 then exit;
                       dau:=0;
                       While (i<length(s))and(s[i+1] in
['a'..'z','A'..'Z','0'..'9']) do inc(i);
                     end;
   '0'..'9'       : begin
                    If dau=0 then exit;
                    dau:=0;
                   while (i<length(s))and(s[i+1] in ['0'..'9']) do  inc(i);
                    end;
         End;
    Inc(i);
  end;
 If (dem=0)and(dau=0) then test:=true;
End;

BEGIN
 Clrscr;
 Write(' Nhap xau S: '); Readln(s);
 If test(s) then Write(' Day la mot bieu thuc dai so.')
   Else Write(' Day khong phai la mot bieu thuc dai so.');
 Readln;
END.
```

13. {$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q+,R+,S+,T-,V+,X+}
{$M 16384,0,655360}
Program P30813;
{Xau ki tu la mot bieu thuc dai so, kiem tra tinh dung dan}

```
Uses  Crt;
Const   MN = 200;
          fn = 'P30813.inp';
          gn = 'P30813.out';
Type  mangst = array [1..MN] of string[9];
        mangr  = array [1..MN] of real;
        mangit = array [1..MN] of integer;
Var   tenbien,ten     : mangst;
       giatri,tri,stack: mangr;
       p  : mangit;
       s,hang,bien     : string;
       sb,st,ip,sst,slb: longint;
       so : real;
       sym : char;
       f,g : text;

Procedure Docbien;
Var s1: string;
     i,cho,dd: integer;
     t: real;
Var d: array [1..2] of longint;

Begin
 slb:=0;
 While not Eof(f) do
   begin
         Readln(f,s1);
         i:=1; Inc(slb);
         Tenbien[slb]:='';
         While s1[i]<>'=' do
           begin
                 If s1[i]<>' ' then
Tenbien[slb]:=Tenbien[slb]+UpCase(s1[i]);
                 Inc(i);
           end;
         FillChar(d,SizeOf(d),0);
         cho:=1;
         dd:=1;
         While (i<=length(s1)) do
           begin
                 If s1[i]='-' then dd:=2;
                 If (s1[i] in ['0'..'9','.']) then
```

```
                Case s1[i] of
                    '0'..'9': d[cho]:=d[cho]*10+ord(s1[i])-ord('0');
                     '.'    : cho:=2;
                    End;
              Inc(i);
          end;
        t:=d[2];
        While t>1 do t:=t/10;
        giatri[slb]:=d[1]+t;
        If dd=2 then giatri[slb]:=-giatri[slb];
   end;
 For i:=1 to slb do
   While length(Tenbien[i])<8 do Tenbien[i]:=Tenbien[i]+' ';
End;

Procedure Nhap;
Begin
 Assign(f,fn); Reset(f);
 Readln(f,s);
 Docbien;
 Close(f);
End;

Procedure Loi(i: byte);
Begin
 Case i of
   1: Write('Khong hop li!');
 End;
End;

Procedure Init;
Var i: integer;
Begin
 For i:=1 to length(s) do s[i]:=UpCase(s[i]);
 sb:=1;
 st:=0;
 ip:=0;
 sst:=0;
End;

Procedure Cach;
Begin
```

```
 While s[sb]=' ' do Inc(sb);
End;

Function Getnum: longint;
Var d: array [1..2] of longint; cho: integer; t: real;
Begin
 FillChar(d.SizeOf(d),0);
 cho:=1;
 While (sb<=length(s))and(s[sb] in ['0'..'9','.']) do
  begin
    Case s[sb] of
       '0'..'9': d[cho]:=d[cho]*10+ord(s[sb])-ord('0');
         '.'     : cho:=2;
      End;
    Inc(sb);
  end;
 t:=d[2];
 While t>1 do t:=t/10;
 so:=d[1]+t;
End;

Procedure Getvar;
Begin
 bien:='';
 While s[sb] in ['A'..'Z','0'..'9'] do
   begin bien:=bien+s[sb]; Inc(sb);  end;
 While length(bien)<8 do bien:=bien+' ';
End;

Procedure Getsym;
Begin
 Cach;
 Case s[sb] of
  '('     : begin sym:=s[sb]; Inc(sb); end;
  ')'     : begin sym:=s[sb]; Inc(sb); end;
  '+'     : begin sym:=s[sb]; Inc(sb); end;
  '-'     : begin sym:=s[sb]; Inc(sb); end;
  '*'     : begin sym:=s[sb]; Inc(sb); end;
  '/'     : begin sym:=s[sb]; Inc(sb); end;
  '%'     : begin sym:=s[sb]; Inc(sb); end;
  '0'..'9': begin sym:='0';   Getnum;  end;
  'A'..'Z': begin sym:='A';   Getvar;  end;
```

```
  Else sym:=#0;
 End;
End;

Function Findso: integer;
Var i: integer;
Begin
 For i:=1 to st do
  If ten[i]='        ' then
   If tri[i]=so then
      begin
            Findso:=i;
            exit;
      end;
 Inc(st);
 Ten[st]:='
 Tri[st]:=so;
 Findso:=st;
End;

Function Findbien: integer;
Var i: integer;
Begin
 For i:=1 to st do
  If pos(bien,ten[i])>0 then begin Findbien:=i; exit; end;
 Inc(st);
 Ten[st]:='?'+bien;
 Findbien:=st;
End;

Function Finddau(c: char): integer;
Begin
 Case c of
  '+': finddau:=-1;
  '-': finddau:=-2;
  '*': finddau:=-3;
  '/': finddau:=-4;
  '%': finddau:=-5;
  '#': finddau:=-6;
 End;
End;
```

```
Procedure Bieuthuc;
Var i: integer; daubt: char;

 Procedure Hangtu;
 Var i: integer;

  Procedure Nhantu;
  Var i: integer;
  Begin
   While sym in ['0','A','('] do
   begin
    Case sym of
       '0': begin i:=findso;   Inc(ip); p[ip]:=i; end;
       'A': begin i:=findbien; Inc(ip); p[ip]:=i; end;
       '(': begin
              getsym; Bieuthuc;
              If sym<>')' then Loi(1);
            end
      End;
    getsym;
   end;
  End;

 Begin
  Nhantu;
  While sym in ['*','/','%'] do
    begin
         i:=finddau(sym);
        getsym;
        Nhantu; Inc(ip); p[ip]:=i;
   end;
 End;

Begin
 If sym in ['+','-'] then begin daubt:=sym; getsym; end
  Else daubt:='+';
 Hangtu;
 If daubt='-' then
    begin
         i:=finddau('#');
         Inc(ip);
         p[ip]:=i;
```

```
    end;
 While sym in ['+','-'] do
   begin
        daubt:=sym;
        getsym;
        Hangtu;
        i:=finddau(daubt);
        Inc(ip);
        p[ip]:=i;
   end;
End;

Procedure Tinh(i: integer);
Begin
 Case p[i] of
  -1: begin stack[sst-1]:=stack[sst-1]+stack[sst]; Dec(sst); end;
  -2: begin stack[sst-1]:=stack[sst-1]-stack[sst]; Dec(sst); end;
  -3: begin stack[sst-1]:=stack[sst-1]*stack[sst]; Dec(sst); end;
  -4: begin stack[sst-1]:=stack[sst-1]/stack[sst]; Dec(sst); end;
  -6: stack[sst]:=-stack[sst];
 End;
End:

Procedure Napso(i: integer);
Begin
 Inc(sst);
 stack[sst]:=tri[p[i]];
End;

Procedure Hoiso(i: integer);
Var j: integer;
Begin
 Delete(ten[p[i]],1,1);
 tri[p[i]]:=0;
 For j:=1 to slb do
  If tenbien[j]=ten[p[i]] then
    begin tri[p[i]]:=giatri[j]; break; end;
 Insert('&',ten[p[i]],1);
End;

Procedure Tinhtoan;
Var i: integer;
```

```
Begin
 For i:=1 to ip do
  If p[i]<0 then Tinh(i)
   Else  Case ten[p[i],1] of
              ' ': Napso(i);
              '?': begin Hoiso(i); Napso(i); end;
              '&': Napso(i);
            End;
End;

Procedure Viet;
Var i: integer;
Begin
 Assign(g,gn); Rewrite(g);
 Writeln(g,s,' = ',stack[1]:4:2);
 Close(g);
End;

BEGIN
 Clrscr;
 Nhap;
 Init;
 Getsym;
 Bieuthuc;
 Tinhtoan;
 Viet;
END.
```

15. Program P30815;

```
{Xoa mot so trong tap hop so de no thanh so tot}
Uses  Crt;
Const   MN = 5002;
         fn = 'P30814.inp';
         gn = 'P30814.out';
Type  mang1  = array [-1..MN] of byte;
      mang1i = array [-1..MN] of integer;
Var   so : mang1;
      tl,pl,tc,pc: mang1i;
      soxoa,sl : integer;
      f,g : text;

Procedure Nhap;
```

```
Var s: string; i: integer;
Begin
 Assign(f,fn); Reset(f);
 sl:=0;
 While not Eof(f) do
   begin
       Readln(f,s);
       For i:=1 to length(s) do
         begin
             Inc(sl);
             so[sl]:=ord(s[i])-ord('0');
         end;
   end;
 Close(f);
End;

Procedure Init;
Begin
 FillChar(tl,SizeOf(tl),0);
 tc:=tl;
 pc:=tl;
 pl:=tl;
End;

Procedure Traiphai;
Var i: integer;
Begin
 For i:=1 to sl do
   begin
       tl[i]:=tl[i-2];
       tc[i]:=tc[i-2];
       If so[i] mod 2=1 then Inc(tl[i]) Else Inc(tc[i]);
   end;
End;

Procedure Phaitrai;
Var i: integer;
Begin
 For i:=sl downto 1 do
   begin
       pl[i]:=pl[i+2];
       pc[i]:=pc[i+2];
```

```
        If so[i] mod 2=1 then Inc(pl[i])
          Else Inc(pc[i]);
  end;
End;
Procedure Tim;
Var  i: integer;
Begin
 soxoa:=0;
 For i:=1 to sl do
   If tc[i-1]+pc[i+2]=tc[i-2]+pc[i+1] then
     If tl[i-1]+pl[i+2]=tl[i-2]+pl[i+1] then
        begin soxoa:=i; exit; end;
End;

Procedure Xuly;
Begin
 Traiphai;
  Phaitrai;
 Tim;
End;

Procedure Viet;
Var i: integer;
Begin
 Assign(g,gn); Rewrite(g);
 If Soxoa=0 then Write(g,'vo nghiem')
   Else
     begin
         For i:=1 to sl do
            begin
                 Write(g,so[i]);
                 If i mod 60=0 then Writeln(g);
             end;
         Writeln(g);
         Writeln(g,'Xoa so thu : ',soxoa);
         For i:=1 to soxoa-1 do
            begin
                  Write(g,so[i]);
                 If i mod 60=0 then Writeln(g);
             end;
         For i:=Soxoa+1 to sl do
             begin
```

```
                Write(g,so[i]);
                If i mod 60=0 then Writeln(g);
            end;
    end;
 Close(g);
End;

BEGIN
 Clrscr;
 Nhap;
 Init;
 Xuly;
 Viet;
END.
```

17. {$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+}

```
{$M 64384,0,655360}
Program P30817;  {Xu li xau}
Uses  Crt;
Const   MN   = 50;
        fn   = 'P30817.inp';
        gn   = 'P30817.out';
        khac = #255;
Type  mangIc = array [1..MN] of char;
      mangIi = array [1..MN] of integer;
      xc  = string[MN];
Var   p : mangIc;
      gt,tu,den   : mangIi;
      s : string;
      si,pi,so : integer;
      sym,chu  : char;
      tg,tgI,sc,sr: xc;
      loai : byte;
      f,g : text;

Procedure Loi;
Begin
End;
Function Dung(sc: xc): xc;
Var i: word; t: string;
```

```
Begin
 dung:='';
 For i:=1 to length(sc) do
    If sc[i]<>sc[1] then exit;
 If length(sc)=1 then Dung:=sc
   Else begin str(length(sc),t); Dung:=sc[1]+t; end;
End;

Function Laplai(x: xc; j: byte): byte;
Var t: byte; cc,cc1: xc;
Begin
 t:=1;
 cc:=copy(x,length(x)-j+1,j);
 While t<=length(x) div j do
   begin
        Inc(t);
        cc1:=copy(x,length(x)-j*t+1,j);
        If cc1<>cc then break;
   end;
 If t>2 then Laplai:=t-1 Else Laplai:=0;
End;

Function Toiuu(sc: xc): xc;
Var s: array [-3..MN] of xc; i,j,t: byte;
Begin
 s[0]:='';
 If Dung(sc)<>'' then begin Toiuu:=Dung(sc); exit; end;
 For i:=1 to length(sc) do
   begin
       s[-1]:=copy(sc,1,i); s[i]:=s[i-1]+sc[i];
       For j:=1 to i div 2 do
         begin
          t:=Laplai(s[-1],j);
          If t<>0 then
           begin
            s[-2]:=Copy(s[-1],i-j+1,j);
             s[-3]:=Toiuu(s[-2]);
             str(t,tg1);
              If length(s[-3])<>1 then tg:=s[i-j*t]+'('+s[-3]+')'+tg1
                          Else tg:=s[i-j*t]+s[-3]+tg1;
                       If length(tg)<=length(s[i]) then s[i]:=tg;
                  end;
```

```
        end;
    end;
 Toiuu:=s[length(sc)];
End;

Procedure Viet2;
Begin
 Writeln(g,sr);
End;

Procedure Xuly2;
Begin
 sr:=Toiuu(sc);
 Viet2;
End;

Procedure Phanloai;
Var i: integer;
Begin
 Loai:=1;
 For i:=1 to length(sc) do
   If sc[i] in ['(',')','0'..'9'] then exit;
 loai:=2;
End;

Procedure Init;
Begin
 si:=1;
 pi:=0;
 s:=s+khac;
End;

Procedure Cach;
Begin
 While s[si]=' ' do Inc(si);
End;
Procedure Getsym;
Begin
 Cach;
 Case s[si] of
   'a'..'z','A'..'Z': begin sym:='B'; chu:=s[si]; Inc(si); end;
   '(',')'          : begin sym:=s[si]; Inc(si); end;
```

```
  Else sym:=khac;
 End;
End;
Procedure Docso;
Begin
 Cach; so:=0;
 While s[si] in ['0'..'9'] do
   begin
        so:=so*10+ord(s[si])-ord('0');
        Inc(si);
   end;
 If so=0 then so:=1;
End;
Procedure Nhapchu;
Begin
 Inc(pi);
 p[pi]:=chu;
 gt[pi]:=so;
End;
Procedure Nhapso(x: integer);
Begin
 Inc(pi);
 p[pi]:=' ';
 gt[pi]:=so;
 tu[pi]:=x;
 den[pi]:=pi-1;
End;

Procedure Lenh;
Begin
 Docso;
 Nhapchu;
 Getsym;
End;
Procedure Daylenh;
Var bd: integer;
Begin
 While sym in ['(','B'] do
  If sym='(' then
    begin
       bd:=pi+1;
       Getsym;
```

```
      Daylenh;
      If sym<>')' then Loi Else Docso;
      Nhapso(bd);
      Getsym;
   end
  Else Lenh;
End;
Procedure Xem;
Var i: integer;
Begin
 For i:=1 to pi do
   Writeln(p[i],' ',gt[i],' ',tu[i],' ',den[i]);
End;
Procedure Viet(x,y: integer);
Var i,j: integer;
Begin
 For i:=x to y do
   If p[i]<>' ' then For j:=1 to gt[i] do Write(g,p[i])
     Else For j:=1 to gt[i]-1 do Viet(tu[i],den[i]);
End;
Procedure Xuly1;
Begin
 s:=sc;
 Init;
 Getsym;
 Daylenh;
 Viet(1,pi);
 Writeln(g);
End;

BEGIN
 Clrscr;
 Assign(f,fn); Reset(f);
 Assign(g,gn); Rewrite(g);
 While not Eof(f) do
   begin
        Readln(f,sc);
        Phanloai;
        If Loai=1 then Xuly1 Else Xuly2;
   end;
 Close(g); Close(f);
END.
```

Bài 9. Vật lí và sinh học

1. Program P30901; {Tim trong tam cua he va khoang cach toi tung diem}

```
Uses Crt;
Const fi = 'INPUT.DAT';
Var a : array [1..4,1..10] of real;
     t,d: real;
     i,j: byte;
Procedure Input;
Var i,j: byte; f: text;
Begin
Assign(f,fi); Reset(f);
For i:=1 to 4 do
  begin
       For j:=1 to 10 do Read(f,a[i,j]);
       Readln(f);
  end;
Close(f);
End;
BEGIN
Clrscr;
Input;
d:=1;
For i:=1 to 10 do d:=d+a[4,i];
Writeln('Trong tam cua he co toa do: ');
For j:=1 to 3 do
 begin
       t:=1;
       For i:=1 to 10 do t:=t+a[j,i]*a[4,i];
       Writeln(' ',t/d:0:4);
 end;
Readln;
END.
```

3. Program P30903; {Qui dao cua con cho khi duoi con meo}

```
Uses  Crt,Graph;
Const   vc=10; vm=5; AO=100; xOA=3*pi/4;
         X0=280; Y0=400; W=3;
Var   xc,xm,yc,ym,T,dx,dy,ds: real;
      Gd,Gm,X,Y : integer;
BEGIN
 Clrscr;
```

```
Write(' T = '); Readln(T);
Gd:=Detect;
InitGraph(Gd,Gm,' ');
xm:=0; ym:=0;
xc:=AO*cos(xOA); yc:=AO*sin(xOA);
X:=Round(xm*W+X0); Y:=Round(Y0-ym*W); PutPixel(X,Y,5);
X:=Round(xc*W+X0); Y:=Round(Y0-yc*W); PutPixel(X,Y,4);
Repeat
  dx:=xm-xc; dy:=ym-yc; ds:=Sqrt(Sqr(dx)+Sqr(dy));
  xc:=xc+vc*T*dx/ds; yc:=yc+vc*T*dy/ds; xm:=xm+vm*t;
  Delay(Round(1000*T));
  X:=Round(xm*W+X0); Y:=Round(Y0-ym*W); PutPixel(X,Y,5);
  X:=Round(xc*W+X0); Y:=Round(Y0-yc*W); PutPixel(X,Y,4);
 Until (xc>=xm)or KeyPressed;
 Readln;
 CloseGraph;
END.
```

5. Program P30905;{Truyen nhiem}

```
Uses Crt;
Const   fia = 'A.TXT';
        fib = 'B.TXT';
        fo = 'C.TXT';
Label 1;
Var a,b,c:Array[1..100,1..100] of byte;
     k,l,m:byte;
     f:text;
     i,j,t:byte;
Begin
  Assign(f,fia);
  Reset(f);
  Readln(f,k,l);
  For i:=1 to k do
    begin
        For j:=1 to l do Read(f,a[i,j]);
        Readln(f);
    end;
  Close(f);
  Assign(f,fib);
  Reset(f);
  Readln(f,l,m);
  For i:=1 to l do
```

```
  begin
      For j:=1 to m do Read(f,b[i,j]);
      Readln(f);
   end;
 Close(f);
 Assign(f,fo);
 ReWrite(f);
 Writeln(f,k,' ',m);
 For i:=1 to k do
  begin
     For j:=1 to m do
      begin
         For t:=1 to l do
          If (a[i,t]=1) and (b[t,j]=1) then
             begin
                  Write(f,' 1');
                  Goto 1;
               end;
         Write(f,' 0');
         1:
      end;
     Writeln(f);
  end;
 Close(f);
END.
```

```
7. {$R+}
Program P30907; {Tinh dien tro tren mach}
Uses Crt;
Const   max=1000; {Toi da co 1000 dien tro trong mach}
Var  f : text;
     m,n,i,j,s1,s2: integer;
     a,b,bac : array [1..max] of integer; {2 dau cua dien tro thu i}
     r : array [1..max] of real; {Dien tro thu i}
     found : boolean;
Procedure Nhap;
Var f: text;
Begin
  Assign(f,'P30914.inp'); Reset(f);
  Readln(f,m);
  n:=0;
  While not EoF(f) do
```

```
    begin
        Inc(n);
        Readln(f,a[n],b[n],r[n]);
    end;
  Close(f);
End;
Procedure Remove(i: integer);
{Xoa dien tro thu i trong mach}
Var j: integer;
Begin
  For j:=i to n-1 do
    begin a[j]:=a[j+1];b[j]:=b[j+1];r[j]:=r[j+1]; end;
  Dec(n);
End;
Procedure Remove2(i,j: integer);
{Xoa 2 dien tro i va j trong mach}
Begin
  If i>j then begin remove(i); remove(j); end
    Else begin remove(j); remove(i); end;
End;
Procedure Add(aa,bb: integer; rr: real);
Begin
  Inc(n);
  a[n]:=aa;
  b[n]:=bb;
  r[n]:=rr;
End;
BEGIN
  Clrscr;
  Nhap;{Init}
  Fillchar(bac,Sizeof(bac),0);
  For i:=1 to n do
    begin
        Inc(bac[a[i]]);
        Inc(bac[b[i]]);
    end;
  Repeat
  {Tim 2 dien tro noi tiep}
```

```
  found:=false;
  For i:=2 to m-1 do
   If bac[i]=2 then
    begin
        For j:=1 to n do
          begin
               If b[j]=i then s1:=j;
               If a[j]=i then s2:=j;
          end;
        Add(a[s1],b[s2],r[s1]+r[s2]);
        Remove2(s1,s2);
        bac[i]:=0;
        found:=true;
        break;
    end;
  If not found then
 {Tim 2 dien tro song song}
    For i:=1 to n-1 do
     begin
        For j:=i+1 to n do
          If (a[i]=a[j])and(b[i]=b[j]) then
            begin
                 Add(a[i],b[i],(r[i]*r[j])/(r[i]+r[j]));
                 Dec(bac[a[i]]);Dec(bac[b[i]]);
                 Remove2(i,j);
                  found:=true;
                  break;
            end;
        If found then break;
    end;
 Until n=1;
 Writeln('Dien tro toan mach la: ',r[1]:0:3);
 Readln;
END.
```

9. Program P30909; {Mo phong chyen dong 2 qua cau}

```
Uses Crt,Graph;
Var x1,x2,v1,v2,t,d,x11,x21: real;
    Gd,Gm : integer;
BEGIN
 Clrscr;
 Write(' X1 = '); Readln(x1);
```

```
  Write(' X2 = '); Readln(x2);
  Write(' V1 = '); Readln(v1);
  Write(' V2 = '); Readln(v2);
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  t:=0.05;
  Repeat
    Circle(Round(x1),240,10); Circle(Round(x2),240,10);
    x11:=x1; x21:=x2;
    x1:=x1+t*v1; x2:=x2+t*v2;
    If Abs(x1-x2)<20 then begin d:=v1; v1:=v2; v2:=d; end;
    SetColor(0);
    Circle(Round(x11),240,10); Circle(Round(x21),240,10);
    Setcolor(15);
  Until Abs(x1-x2)>1000;
  CloseGraph;
END.
```

11. Program P30911; {Mo phong chyen dong cua n qua cau}

```
Uses Crt,Graph;
Const   fi = 'INPUT.DAT';
Var   t,d : real;
      f : text;
      a : array [1..100,1..3] of real;
      Gd,Gm,n,i,j: integer;
BEGIN
  Assign(f,fi); Reset(f);
  Readln(f,n);
  For i:=1 to n do Read(f,a[i,1]); Readln(f);
  For i:=1 to n do Read(f,a[i,2]); Readln(f);
  Close(f);
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  t:=0.05;
  Repeat
    For i:=1 to n do
      begin
          Circle(Round(a[i,1]),240,10);
          a[i,3]:=a[i,1];
          a[i,1]:=a[i,1]+t*a[i,2];
      end;
    For i:=1 to n-1 do
```

```
    For j:=i+1 to n do
      If Abs(a[i,1]-a[j,1])<20 then
        begin
              d:=a[i,2];
              a[i,2]:=a[j,2];.
              a[j,2]:=d;
        end;
   SetColor(0);
   For i:=1 to n do Circle(Round(a[i,3]),240,10);
   Setcolor(15);
 Until KeyPressed;
 CloseGraph;
END.
```

13. Program P30913; {Mo phong chyen dong cua N qua cau}

```
Uses Crt,Graph;
Const   r=10;
Var   n,i,x1,y1,x2,y2,Gd,Gm: integer;
      x,y,vx,vy: array [1..100] of integer;
Procedure Nhap;
Var   f: text; i: integer;
Begin
  Assign(f,'P30913.inp'); Reset(f);
  Readln(f,x1,y1,x2,y2);
  Readln(f,n);
  For i:=1 to n do Readln(f,x[i],y[i]);
  For i:=1 to n do Readln(f,vx[i],vy[i]);
  Close(f);
End;
BEGIN
  Nhap;
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  Rectangle(x1,y1,x2,y2);
  Repeat
    Setcolor(9);
    For i:=1 to n do Circle(x[i],y[i],r);
    Delay(10);
    SetColor(0);
```

```
  For i:=1 to n do Circle(x[i],y[i],r);
  For i:=1 to n do
    begin
          x[i]:=x[i]+vx[i]; y[i]:=y[i]+vy[i];
          If (x[i]-x1<=r+2)or(x2-x[i]<=r+2) then vx[i]:=-vx[i];
          If (y[i]-y1<=r+2)or(y2-y[i]<=r+2) then vy[i]:=-vy[i];
     end;
 Until KeyPressed;
 CloseGraph;
END.
```

15. Program P30915; {Tim toa do trong tam cua chat the}

```
Uses Crt;
Var f : text;
     i,n : integer;
     x,y,m : array [1..100] of real;
     xo,yo,Sm,Smx,Smy: real;
Procedure Nhap;
Var f: text;
Begin
 Assign(f,'P30915.inp'); Reset(f);
 n:=0;
 While not EOF(f) do
  begin
        Inc(n);
        Readln(f,x[n],y[n],m[n]);
   end;
 Close(f);
End;
BEGIN
 Clrscr;
 Nhap;
 Sm:=0; Smx:=0; Smy:=0;
 For i:=1 to n do Sm:=Sm+m[i];
 For i:=1 to n do Smx:=Smx+m[i]*x[i];
 For i:=1 to n do Smy:=Smy+m[i]*y[i];
 xo:=Smx/Sm; yo:=Smy/Sm;
 Writeln('Toa do trong tam : (',xo:0:2,',',yo:0:2,')');
 Readln;
END.
```

Bài 10. Âm nhạc và hội hoạ

1. Program P31001; {Day not dung trong am nhac}

```
Uses Crt;
Const d3=261;
Var  a: array [0..95] of integer;
     i: integer;
     q,f: real;

BEGIN
 q:=Exp(ln(2)/12);
 f:=d3;
 For i:=36 to 95 do
  begin a[i]:=round(f); f:=f*q;  end;
 f:=d3;
 For i:=36 downto 0 do
   begin  a[i]:=round(f); f:=f/q;  end;
 For i:=0 to 95 do
    begin
         Sound(a[i]);
         Delay(100);
    end;
 NoSound;
END.
```

3. Gam mi trưởng: Mi (E), Fa thăng (FF), Sol thăng (GF), La (A), Si (B), Đô thăng (CF), Rê thăng (DF).

Gam sol trưởng: Sol (G), La (A), Si (B), Đô (C), Rê (D), Mi (E), Fa thăng (FF).

```
Program P31003;   {Gam truong}
Uses Crt;
Const   Major: array [0..7] of byte=(0,2,4,5,7,9,11,12);
        Notes: array [0..11] of string =
('C','CF','D','DF','E','F','FF','G','GF','A','AF','B');
        d3 = 261;
Var  Gam: string;
     a: array [0..95] of integer;
     i0,i: integer;
     q,f: real;
```

```
BEGIN
 q:=Exp(ln(2)/12);
 f:=d3;
 For i:=36 to 95 do
   begin  a[i]:=round(f); f:=f*q;  end;
 f:=d3;
 For i:=36 downto 0 do
   begin  a[i]:=round(f); f:=f/q;  end;
 Write('Nhap ten gam (C/CF/D/DF/E/F/FF/G/GF/A/AF/B): ');
Readln(Gam);
 For i:=0 to 11 do
   If Gam=Notes[i] then i0:=i+12*3;
 For i:=0 to 7 do  begin Sound(a[i0+Major[i]]); Delay(500); end;
 For i:=7 downto 0 do begin Sound(a[i0+Major[i]]); Delay(500); end;
 Nosound;
END.
```

5. Program P31005; {Gam thu}

```
Uses  Crt;
Const   Minor: array [0..7] of byte=(0,2,3,5,7,8,10,12);
          Notes: array [0..11] of string
=('C','CF','D','DF','E','F','FF','G','GF','A','AF','B');
          d3 = 261;
Var   Gam: string;
        a: array [0..95] of integer;
        i0,i: integer;
        q,f: real;

BEGIN
 q:=Exp(ln(2)/12);
 f:=d3;
 For i:=36 to 95 do
   begin  a[i]:=round(f); f:=f*q; end;
 f:=d3;
 For i:=36 downto 0 do
   begin  a[i]:=round(f); f:=f/q; end;
 Write('Nhap ten gam (C/CF/D/DF/E/F/FF/G/GF/A/AF/B): ');
Readln(Gam);
 For i:=0 to 11 do
   If Gam=Notes[i] then i0:=i+12*3;
 For i:=0 to 7 do begin Sound(a[i0+Minor[i]]); Delay(500); end;
```

```
 For i:=7 downto 0 do begin Sound(a[i0+Minor[i]]); Delay(500); end;
 Nosound;
END.
```

7.
```
Program P31007;  {Bai hat 'Cho con' cua Pham Trong Cau}
Uses  Crt;
Const   Notes: array [0..12] of string
=('L','C','CF','D','DF','E','F','FF','G','GF','A','AF','B');
        a0=32.625;
        T=1000; D=500; MD=250;
Var   f: text;
      S,N: string;
      O,Du,Code: integer;

Procedure Play(Octave: integer; Note: string; Duration: integer);
Var Fr:real; i:Integer;
Begin
 Fr:=a0;
 For i:=1 to Octave do Fr:=Fr*2;
 i:=0;
 While Notes[i]<>Note do begin Fr:=Fr*Exp(ln(2)/12); inc(i); end;
 If Note<>'L' then
   begin
      Sound(Round(Fr));
      Delay(Duration);
      NoSound;
      Delay(50);
   end
  Else begin Nosound; Delay(Duration+50); end;
End;

BEGIN
 Assign(f,'P31007.INP'); Reset(f);
 Repeat
  Read(f,O);
  Readln(f,S);
  While S[1]=' ' do Delete(S,1,1);
  While S[length(S)]=' ' do Delete(S,length(S),1);
  N:=Copy(S,1,Pos(' ',s)-1);
  Delete(S,1,Pos(' ',s));
  While S[1]=' ' do Delete(S,1,1);
  If S='T' then Du:=T;
```

```
  If S='D' then Du:=D;
  If S='MD' then Du:=MD;
  Play(O,N,Du);
 Until EoF(f);
 Close(f);
END.
```

9. Program P31009; {Doan ten bai hat va ten tac gia}

```
Uses Crt;
Const  Notes: array [0..12] of string
=('L','C','CF','D','DF','E','F','FF','G','GF','A','AF','B');
        a0=32.625;
        T=1000; D=500; MD=250;
Var  f: text;
     S,N,Baihat: string;
     O,Du,Code,m: integer;
     c: char;

Procedure Play(Octave: integer; Note: string; Duration: integer);
Var Fr:real; i:Integer;
Begin
 Fr:=a0;
 For i:=1 to Octave do Fr:=Fr*2;
 i:=0;
 While Notes[i]<>Note do begin Fr:=Fr*Exp(ln(2)/12); inc(i); end;
 If Note<>'L' then
   begin
      Sound(Round(Fr));
      Delay(Duration);
      NoSound;
      Delay(50);
   end
  Else begin Nosound; Delay(Duration+50); end;
End;

BEGIN
 Repeat
   ClrScr;
   Writeln('Ban hay nghe bai hat sau:');
   Randomize;
   m:=Random(3);
   Case m of
```

```
    1: Baihat:='P310081.Inp';
    2: Baihat:='P310082.Inp';
    3: Baihat:='P310083.Inp';
  End;
  Assign(f,'P310N\'+Baihat);
  Reset(f);
  Repeat
    Read(f,O);
    Readln(f,S);
    While S[1]=' ' do Delete(S,1,1);
    While S[length(S)]=' ' do Delete(S,length(S),1);
    N:=Copy(S,1,Pos(' ',s)-1);
    Delete(S,1,Pos(' ',s));
    While S[1]=' ' do Delete(S,1,1);
    If S='T' then Du:=T;
    If S='D' then Du:=D;
    If S='MD' then Du:=MD;
    Play(O,N,Du);
  Until EoF(f);
  Close(f);
  Writeln('Bai hat ban vua nghe la bai hat: ');
  Writeln('1. "Nam ngon tay" cua Tran Van Thu.');
  Writeln('2. "Doi ken ti hon" cua Phan Huynh Dieu.');
  Writeln('3. "Cho con" cua Pham Trong Cau.');
  Writeln('4. "Co va me" cua Pham Tuyen.');
  c:=ReadKey;
  If m=Ord(c)-48 then Writeln('Dung!')
    Else Writeln('Sai!');
  Write('Ban co doan tiep khong? (c/k): ');
  Repeat c:=ReadKey; c:=Upcase(c); Until c in ['C','K'];
 Until c='K';
END.
```

11. Program P31011; {Doan ten bai hat nguoc}

```
Uses Crt;
Const  Notes: array [0..12] of string
=('L','C','CF','D','DF','E','F','FF','G','GF','A','AF','B');
        C0=32.625;
        T=1000; D=500; MD=250;
Var  f: text;
     S,N,Baihat: string;
     O,Du,Code,m: integer;
```

```
    c: char;

Procedure Play(Octave: integer; Note: string; Duration: integer);
Var Fr:real; i:Integer;
Begin
 Fr:=C0;
 For i:=1 to Octave do Fr:=Fr*2;
 i:=0;
 While Notes[i]<>Note do begin Fr:=Fr*Exp(ln(2)/12); inc(i); end;
 If Note<>'L' then
    begin
          Sound(Round(Fr));
          Delay(Duration);
          NoSound:
          Delay(50);
    end
   Else begin Nosound; Delay(Duration+50); end;
End;

Procedure Hat1(Baihat: String);
Begin
 Assign(f,'P310N\'+Baihat);
 Reset(f);
 Repeat
   Read(f,O);
   Readln(f,S);
   While S[1]=' ' do Delete(S,1,1);
   While S[length(S)]=' ' do Delete(S,length(S),1);
   N:=Copy(S,1,Pos(' ',s)-1);
   Delete(S,1,Pos(' ',s));
   While S[1]=' ' do Delete(S,1,1);
   If S='T' then Du:=T;
   If S='D' then Du:=D;
   If S='MD' then Du:=MD;
   Play(O,N,Du);
 Until EoF(f);
 Close(f);
End;

Procedure Hat2(m: integer);
Var N: array [1..100] of string[2];
     O,Du: array [1..100] of integer;
```

```
    i,k: integer;
Begin
 Case m of
  0: Baihat:='P310081.Inp';
  1: Baihat:='P310082.Inp';
  2: Baihat:='P310083.Inp';
 End;
 Assign(f,'P310N\'+Baihat); Reset(f);
 k:=0;
 Repeat
  Inc(k);
  Read(f,O[k]);
  Readln(f,S);
  While S[1]=' ' do Delete(S,1,1);
  While S[length(S)]=' ' do Delete(S,length(S),1);
  N[k]:=Copy(S,1,Pos(' ',s)-1);
  Delete(S,1,Pos(' ',s));
  While S[1]=' ' do Delete(S,1,1);
  If S='T' then Du[k]:=T;
  If S='D' then Du[k]:=D;
  If S='MD' then Du[k]:=MD;
  Writeln(k:3,O[k]:5,N[k]:5,Du[k]:5);
 Until EoF(f);
 Close(f);
 For i:=k downto 1 do Play(O[i],N[i],Du[i]);
End;

BEGIN
 ClrScr;
 Writeln('Ban hay nghe 3 bai hat sau:');
 Writeln('1. "Nam ngon tay" cua Tran Van Thu.');
 Writeln('2. "Doi ken ti hon" cua Phan Huynh Dieu.');
 Writeln('3. "Cho con" cua Pham Trong Cau.');
 Hat1('P310081.Inp');
 Hat1('P310082.Inp');
 Hat1('P310083.Inp');
 Repeat
  ClrScr;
  Writeln('Ban hay nghe bai hat nguoc:');
  Randomize;
  m:=Random(3);
  Hat2(m);
```

```
  Writeln('Bai hat ban vua nghe la bai hat: ');
  Writeln('1. "Nam ngon tay" cua Tran Van Thu.');
  Writeln('2. "Doi ken ti hon" cua Phan Huynh Dieu.');
  Writeln('3. "Cho con" cua Pham Trong Cau.');
  c:=ReadKey;
  If m=Ord(c)-49 then Writeln('Dung!')
   Else Writeln('Sai!');
  Write('Ban co doan tiep khong? (c/k): ');
  c:=ReadKey; c:=Upcase(c);
 Until c='K';
END.
```

13. Program P31013; {Ghi nho ban nhac tu Piano}

```
Uses  Crt,Dos;
Const   NoteSet: set of char=['Q','2','W','3','E','R','5','T','6',
'Y','7','U','I','9','O','0','P','[','=',']',#13];
        Notes: array [1..20] of string=('Q','2','W','3','E','R','5',
'T','6','Y','7','U','I','9','O','0','P','[','=',']');
Var  c: char;
     i,k: integer;
     N: array [1..100] of string[2];
     Du: array [0..100] of integer;
     h,h1,m,m1,s,s1,hund,hund1 : Word;

Procedure Play(Note: string; Duration: integer);
Var Fr:real; i:Integer;
Begin
 Fr:=261;
 i:=1;
 While (Notes[i]<>Note)and(i<21) do
    begin
        Fr:=Fr*Exp(ln(2)/12);
        inc(i);
    end;
 Sound(Round(Fr));
 Delay(Duration);
 NoSound;
 Delay(50);
End;

BEGIN
 ClrScr;
```

```
 Writeln('Hay choi mot ban nhac!');
 Writeln('Enter: Xong.');
 k:=0;
 Repeat
   h1:=h; m1:=m; s1:=s; hund1:=hund;
   Inc(k);
   Repeat
     c:=ReadKey; If c= #0 then c:=ReadKey;
     c:=Upcase(c);
   Until c in NoteSet;
   N[k]:=C;
   GetTime(h,m,s,hund);
   Du[k-1]:= (((h-h1)*60+m-m1)*60+s-s1)*100+hund-hund1;
 Until c= #13;
 For i:=1 to k-1 do Play(N[i],Du[i]*9);
END.
```

15. Đoạn nhạc được dịch sang chuẩn MIDI như sau:

30 ON 49

40 ON 49

60 ON 44

80 ON 46

110 ON 44

120 ON 44

130 ON 46

140 ON 41

160 ON 41

190 ON 44

200 ON 41

220 ON 41

240 ON 44

260 ON 44
280 ON 39
300 ON 39
320 ON 37
340 OF 37

350 ON 41
360 ON 44
380 ON 46
400 ON 49

430 ON 46
440 ON 46
450 ON 49
460 ON 41
480 ON 44

510 ON 46
520 ON 44
540 ON 51
560 ON 51
570 ON 49

580 ON 49
590 ON 51
600 ON 44
620 ON 51
640 ON 49
660 OF 49

Bạn có thể ghi nội dung trên vào text file và dùng chương trình ở bài 16 để kiểm tra.

```
16. Program P31016;  {Doc bai hat theo chuan MIDI}
Uses  Crt;
Const  C0 =32.625;
Var   f: text;
      S: string;
      k,i,Code: integer;
      a: array [1..3,1..500] of integer;

Procedure Nhap;
Begin
 Assign(f,'P31017.out'); Reset(f);
 k:=0;
 Repeat
    Inc(k);
    Read(f,a[1,k]);
    Readln(f,S);
    While S[1]=' ' do Delete(S,1,1);
    While S[length(S)]=' ' do Delete(S,length(S),1);
    If Copy(S,1,Pos(' ',S)-1)='ON' then a[2,k]:=1
      Else a[2,k]:=0;
    Delete(S,1,Pos(' ',S));
    While S[1]=' ' do Delete(S,1,1);
    Val(S,a[3,k],Code);
 Until EoF(f);
 Close(f);
End;

Procedure Sapxep; {Sap xep theo A[1,i]}
Var i,j,tg: integer;
Begin
 For i:=1 to k-1 do
   For j:=i+1 to k do
    If a[1,i]>a[1,j] then
      begin
          tg:=a[1,i]; a[1,i]:=a[1,j]; a[1,j]:=tg;
          tg:=a[2,i]; a[2,i]:=a[2,j]; a[2,j]:=tg;
          tg:=a[3,i]; a[3,i]:=a[3,j]; a[3,j]:=tg;
      end;
```

```
End;

Procedure Play(N: integer);
Var Fr: real; i,ON:Integer;
Begin
 Fr:=C0;
 For i:=1 to a[3,N] do Fr:=Fr*Exp(ln(2)/12);
 If a[2,N]=1 then
   begin
       ON:=a[3,N];
       Sound(Round(Fr));
       Delay((a[1,N+1]-a[1,N])*30);
   end
 Else
   begin
        If ON=a[2,N] then NoSound;
        Delay((a[1,N+1]-a[1,N])*30)
   end;
 Nosound;
 Delay(30);
End;

BEGIN
 Nhap;
 Sapxep;
 For i:=1 to k-1 do Play(i);
END.
```

17. Program P31017; {Mo phong tieng chuong dien thoai}

```
Uses Crt;
Const   Notes: array [1..12] of string
=('C','CF','D','DF','E','F','FF','G','GF','A','AF','B');
          C0=32.625;

Procedure Play(Octave: integer; Note: string; Duration: integer);
Var Fr:real; i:integer;
Begin
 Fr:=C0;
 For i:=1 to Octave do Fr:=Fr*2;
 i:=1;
 While Notes[i]<>Note do begin Fr:=Fr*Exp(ln(2)/12); inc(i); end;
 Sound(Round(Fr));
```

```
 Delay(Duration);
 NoSound;
 Delay(10);
End;

Procedure SoftAlarm;
Var i:integer;
Begin
 For i:=1 to 8 do
   begin
        Play(4,'G',70);
        Play(4,'C',70);
   end;
 delay(1000);
End;

BEGIN
 Writeln('Press any key to Stop...');
 Repeat SoftAlarm Until KeyPressed;
END.
```

19. Program P31019; {Mo phong tieng coi hu}

```
Uses  Crt;

Procedure Siren;
Var Fr: integer;
Begin
For Fr:=500 to 2000 do
  begin  Delay(1); Sound(Fr); end;
 For Fr:=2000 downto 500 do
    begin  Delay(1); Sound(Fr);  end;
 NoSound;
End;

BEGIN
  Repeat Siren Until KeyPressed;
END.
```

Bài 11. Trò chơi

Chương trình sau được viết để sử dụng cho các bài sau. Giả sử file này được ghi trong thư mục TP của ổ C, và tên file là Mouse.pas:

```
{Mouse.pas}
Var mouseX,mouseY:word;
    mouseL,mouseR:boolean;

Procedure showmouse;
Begin
 Asm
  mov ax,1
  int 33h
 End;
End;

Procedure hidemouse;
Begin
 Asm
  mov ax,2
  int 33h
 End;
End;

Function mouseevent:boolean;
Var t:word;
Begin
 Asm
  mov ax,3
  int 33h
  mov mouseX,cx
  mov mouseY,dx
  mov t,bx
 End;
 mouseY:=mouseY div 8 +1;
 mouseX:=mouseX div 8 +1;
 mouseEvent:=t<>0;
 mouseL:=boolean(t and 1);
 mouseR:=boolean(t and 2);
End;

Function mouseeventgr:boolean;
```

```
Var t:word;
Begin
 Asm
  mov ax,3
  int 33h
  mov mouseX,cx
  mov mouseY,dx
  mov t,bx
 End;
 inc(mouseX);
 inc(mouseY);
 mouseEventgr:=t<>0;
 mouseL:=boolean(t and 1);
 mouseR:=boolean(t and 2);
End;
```

1. Program P31101;{Ban co 7 o}

```
Uses crt;
{$I c:\TP\mouse.pas}
Var a:array[0..7] of byte;
     fc:byte;
Procedure hien;
Var  i:byte;
     s:string;
Begin
 hidemouse;
 For i:=1 to 7 do
   begin
        If i = fc then s:=' X ' else s:=' ';
        Case a[i] of
           1: Textbackground(1);
           2: Textbackground(4);
           0: Textbackground(7);
         End;
        gotoxy(i*6,1);
        write(s);
   End;
  showmouse;
End;

Procedure swap(Var a,b:byte);
```

```
Var tg:byte;
Begin
 tg:=a;
 a:=b;
 b:=tg;
End;

Procedure init;
Var i:byte;
Begin
 a[4]:=0;
 a[0]:=0;
 fc:=3;
 For i:=1 to 3 do
  begin
      a[i]:=1;
      a[8-i]:=2;
  end;
End;

Function test:boolean;
Var i:byte;
Begin
 test:=false;
 For i:=1 to 3 do If a[i]<>2 then exit;
 For i:=5 to 7 do If a[i]<>1 then exit;
 test:=true;
End;

Procedure play;
label  l1;
Var i,t:byte;
    x:word;
Begin
 showmouse;
 hien;
 Repeat
   If mouseevent and mouseL and (mouseY=1) then
     begin
         x:=mouseX;
         t:=0;
         For i:=1 to 7 do
```

```
            If (x>=i*6) and (x<(i+1)*6) then
              Begin
                    t:=i;
                    break;
              End;
          If a[t]<>0 then Begin fc:=t;End;
          If (a[t]= 0)and(t<>0)and(abs(t-fc)<3)then
            Begin
                swap(a[t],a[fc]);
                fc:=t;
            End;
          hien;
      end;
  Until test or keypressed;
  If not test then exit;
  writeln;
  textbackground(0);
  writeln(' Ban da hoan thanh tro choi ');
  readln;
End;

BEGIN
  clrscr;
  init;
  play;
END.
```

3. Program P31103;{xep mang 2x3}

```
Uses crt;
{$I c:\TP\mouse.pas}
Const   px:array[1..4] of shortint = (0,1,0,-1);
        py:array[1..4] of shortint = (-1,0,1,0);
Var a : array[1..2,1..3] of byte;

Procedure hien;
Var i,j:byte;
Begin
  hidemouse;
  For i:=1 to 2 do
    For j:=1 to 3 do
      Begin
          gotoxy(35+j*2,i+10);
```

```
        textcolor(a[i,j]);
        Write('+');        {soạn thảo '+' bằng cách ấn Alt +219 - gõ số ở
bên phải bàn phím}
    End;
  showmouse;
End;

Procedure init;
Begin
  a[1,1]:=2;
  a[1,2]:=12;
  a[1,3]:=1;
  a[2,1]:=0;
  a[2,2]:=15;
  a[2,3]:=4;
  hien;
End;

Procedure swap(Var a,b:byte);
Var tg:byte;
Begin
  tg:=a;
  a:=b;
  b:=tg;
End;

function test:boolean;
Var i:byte;
Begin
  test:=false;
  If a[1,1]<>2 then exit;
  If a[1,2]<>12 then exit;
  If a[2,1]<>0 then exit;
  If a[2,2]<>15 then exit;
  If a[1,3]<>4 then exit;
  If a[2,3]<>1 then exit;
  test:=true;
End;

Procedure play;
Var x,y,i,j,i1,j1,k:word;
Begin
```

```
showmouse;
hien;
repeat
 If mouseevent and mouseL then
   begin
        x:=mouseX;
        y:=mouseY;
        For i:=1 to 4 do
          If (x>=35+i*2) and (x<37+i*2) then break;
        For j:=1 to 3 do
          If (y=10+j) then break;
        If (i in [1..3]) and (j in [1..2]) then
          begin
               For k:=1 to 4 do
                 begin
              i1:=i+px[k];
              j1:=j+py[k];
                 If (i1 in [1..3]) and (j1 in [1..2]) and (a[j1,i1]=15) then
                        begin
                             swap(a[j,i],a[j1,i1]);
                             break;
                        end;
                 end;
          end;
        hien;
   end;
 Until test or keypressed;
 If not test then exit;
 Writeln;
 Writeln(' Ban da hoan thanh tro choi');
 Readln;
End;

BEGIN
 textbackground(7);
 clrscr;
 init;
 play;
END.
```

5. Program P31105;{doi cho cac o vuong mang 4x4}

```
Uses crt;
{$I c\TP\mouse.pas}
Const   px:array[1..4] of shortint = (0,1,0,-1);
        py:array[1..4] of shortint = (-1,0,1,0);
Var a : array[1..4,1..4] of byte;

Procedure hien;
Var i,j:byte;
Begin
  hidemouse;
  For i:=1 to 4 do
    For j:=1 to 4 do
      begin
            gotoxy(30+j*3,i+10);
            textbackground(a[i,j]);
            textcolor(15);
            write(a[i,j]:3);
      end;
  showmouse;
End;

Procedure init;
Var i,j,k:byte;
Begin
  fillchar(a,sizeof(a),0);
  randomize;
  For k:=1 to 15 do
    Begin
      repeat
        i:=random(4)+1;
        j:=random(4)+1;
      until a[i,j]=0;
      a[i,j]:=k;
    End;
End;

Procedure swap(Var a,b:byte);
Var tg:byte;
Begin
```

```
  tg:=a;
  a:=b;
  b:=tg;
End;
function test:boolean;
Var i,j:byte;
Begin
  test:=false;
  For i:=1 to 4 do
    For j:=1 to 4 do
      If ((i<>4) or (j<>4)) and (a[i,j]<>(i-1)*4+j) then exit;
  test:=true;
End;
Procedure play;
Var x,y,i,j,i1,j1,k:word;
Begin
  showmouse;
  hien;
  Repeat
    If mouseevent and mouseL then
      Begin
          x:=mouseX;
          y:=mouseY;
          For i:=1 to 5 do
            If (x>=30+i*3) and (x<33+i*3) then break;
          For j:=1 to 5 do
            If (y=10+j) then break;
          If (i in [1..4]) and (j in [1..4]) then
            Begin
                For k:=1 to 4 do
                  Begin
                    i1:=i+px[k];
                    j1:=j+py[k];
                     If (i1 in [1..4]) and (j1 in [1..4]) and
(a[j1,i1]=0) then
                          begin
                                swap(a[j,i],a[j1,i1]);
                                break;
                           end;
                  End;
```

```
            End;
          hien;
    End;
  Until test or keypressed;
  If not test then exit;
  writeln;
  writeln(' Ban da hoan thanh tro choi');
  readln;
End;

BEGIN
  textbackground(0);
  clrscr;
  init;
  play;
END.
```

7. Program P31107;{ba con ngua}

```
Uses graph,crt;
Type    arr = array[1..500,1..2] of longint;
Var
        n: longint;
        mh,mode: integer;
        v,b:array[1..3] of longint;
        c1,c2,c3: arr;
        t:array[1..3,1..500] of longint;
        count: longint;

Procedure nhap;
Begin
  write('n = ');
  readln(n);
End;
Procedure khoidong;
Begin
  mh:=0;
  initgraph(mh,mode,' ');
End;
Procedure vantoc;
Begin
  v[1]:=random(3)+1;
  v[2]:=random(3)+1;
```

```
  v[3]:=random(3)+1;
End;

Procedure ve;
Begin
  setcolor(magenta);
  setcolor(magenta);
  circle(100,240,90);
  circle(300,240,90);
  circle(500,240,90);
End;

Procedure tinhtd;
Var i,c : longint;
Begin
count:=0;
For i:=100 to 190 do
  Begin
       inc(count);
       c1[count,1]:=i;
       c:=round(sqrt((8100-sqr(i-100))));
       c1[count,2]:=c+240;
       c1[180-count,1]:=i;
       c1[180-count,2]:=-c+240;
       c:=round(sqrt((8100-sqr(c1[180-count,2]-240))));
       c1[180+count,1]:=-c+100;
       c1[180+count,2]:=c1[180-count,2];
       c1[360-count,1]:=c1[180+count,1];
       c1[360-count,2]:=c1[count,2];
   End;
count:=360;
For i:=1 to 360 do
   Begin
          c2[i,1]:=c1[i,1]+200;
          c3[i,1]:=c1[i,1]+400;
          c2[i,2]:=c1[i,2];
          c3[i,2]:=c1[i,2];
   End;
End;

Procedure tron(x,y,mau : longint);
Begin
```

```
 setcolor(mau);
 circle(x,y,3);
End;

Procedure thoigian(k : integer);
Var i : integer;
Begin
For i:=1 to 360 do t[k,i]:=i*v[k];
End;
Procedure xuly;
Var td : integer;
Begin
 vantoc;
 thoigian(1); thoigian(2); thoigian(3);
 b[1]:=2; b[2]:=2; b[3]:=2; td:=1;
 tron(c1[1,1],c2[1,2],green);
 tron(c2[1,1],c2[1,2],red);
 tron(c3[1,1],c2[1,2],blue);
 delay(100);
 repeat
  delay(10);
  inc(td);
  if td=t[1,b[1]] then
   Begin
       tron(c1[b[1]-1,1],c1[b[1]-1,2],white);
       setcolor(magenta);
       circle(100,240,90);
       tron(c1[b[1],1],c1[b[1],2],green);
       inc(b[1]);
   End;
 if td=t[2,b[2]] then
  Begin
       tron(c2[b[2]-1,1],c2[b[2]-1,2],white);
       setcolor(magenta);
       circle(300,240,90);
       tron(c2[b[2],1],c2[b[2],2],red);
       inc(b[2]);
  End;
 if td=t[3,b[3]] then
  Begin
       tron(c3[b[3]-1,1],c3[b[3]-1,2],white);
       setcolor(magenta);
```

```
        circle(500,240,90);
        tron(c3[b[3],1],c3[b[3],2],blue);
        inc(b[3]);
    End;
  Until (b[1]=360) or (b[2]=360) or (b[3]=360);
End;

Procedure ketthuc;
Begin
  closegraph;
  if b[n]=360 then
    Begin
        writeln('Ban thang roi do');
        readln;
        exit;
    End;
  writeln('He he ban thua roi nha');
  readln;
End;

BEGIN
  clrscr;
  randomize;
  nhap;
  khoidong;
  setbkcolor(white);
  ve;
  tinhtd;
  xuly;
  ketthuc;
END.
```

9. Program P31109;{Ban may bay}

```
Uses crt;
Var   plane:array[1..10] of record
                x,y:integer;
                enable:boolean;
                v:longint;
            End;
      bullet:array[1..10] of record
                x,y : integer;
```

```
            enable : boolean;
            v : longint;
          End;
      px : integer;
      show : boolean;
      t : longint;

Procedure hien;
Var i:byte;
Begin
 clrscr;
 For i:=1 to 10 do
   Begin
      If plane[i].enable then
        With plane[i] do
          Begin
               gotoxy(x,y);
               write('Plane')
          End;
      If bullet[i].enable then
        With bullet[i] do
           Begin
                gotoxy(x,y);
                write('^');
           End;
    End;
  gotoxy(px,25);
  write('^');
End;

Procedure short;
Var i:byte;
Begin
  For i:=1 to 10 do
    If bullet[i].enable=false then
      With bullet[i] do
        Begin
            x:=px;
            y:=25;
            enable:=true;
            v:=0;
            show:=true;
```

```
      End;
End;

Procedure newplane;
Var i:byte;
Begin
 For i:=1 to 10 do
  If plane[i].enable=false then
   With plane[i] do
     Begin
          x:=random(80)+1;
          y:=random(10)+1;
          enable:=true;
          v:=0;
          show:=true;
     End;
End;
Procedure init;
Var i:byte;
Begin
 For i:=1 to 10 do bullet[i].enable:=false;
 For i:=1 to 10 do newplane;
 px:=40;
 t:=0;
End;
Procedure test;
Var i,j:byte;
Begin
 For i:=1 to 10 do
  If plane[i].enable then
   For j:=1 to 10 do
    If (bullet[j].enable) and (bullet[j].y=plane[i].y) and
(plane[i].x<=bullet[j].x)

        and (-plane[i].x+bullet[j].x<5) then
            begin
                 bullet[i].enable:=false;
                 plane[i].enable:=false;
            end;
 For i:=1 to 10 do
  If (bullet[i].enable) and (bullet[i].y=1) then bullet[i].enable:=false;
End;
```

```
Procedure update;
Var  i : byte;
     dx,dy : integer;
Begin
 If t<10000 then inc(t)
  else Begin
            t:=0;
            newplane;
       End;
 For i:=1 to 10 do
  Begin
       With bullet[i] do
         If enable then
          Begin
               If v<1000 then inc(v)
                 else Begin
                          v:=0;
                          dec(y);
                          show:=true;
                       End;
          end;
        With plane[i] do
          If enable then
           Begin
               If v<10000 then inc(v)
                 else  Begin
                          v:=0;
                          dx:=random(3)-1;
                          dy:=random(3)-1;
                          If (x+dx>0) and (x+dx<81) then x:=x+dx;
                          If (y+dy>0) and (y+dy<11) then y:=y+dy;
                          show:=true;
                       end;
          end;
  end;
End;
Procedure play;
Var ch:char;
Begin
 ch:= #0;
 repeat
```

```
  If keypressed then
   Begin
       ch:=readkey;
       If ch= #0 then ch:=readkey;
       case ch of
            #75:If px>1 then dec(px);
            #77:If px<80 then inc(px);
            #32:short;
          end;
   end;
  show:=false;
  update;
  test;
  If show then hien;
 Until ch= #27;
End;

BEGIN
 init;
 play;
END.
```

11. Program P31111;{tim so}

```
Uses crt;
Var   b,a,k,dem:word;
      ch:char;
BEGIN
 Clrscr;
 b:=1000;
 a:=0;
 dem:=0;
 Repeat
  inc(dem);
  k:=(a+b) div 2;
  write(dem,' Co lon hon ',k,' khong C/K');readln(ch);
  If upcase(ch)= 'C' then a:=k
    else b:=k;
 Until b-a<2;
 If a<b then
  Begin
      If upcase(ch)='C' then a:=b
        else b:=a;
```

```
  End;
 Write(' So ban nghi la ', a);
 Readln;
END.
```

13. Program P31113;{Boc soi}

```
Uses crt;
Var   d,c,k:integer;
      a:array[1..2] of integer;
      ch:char;
BEGIN
 Write(' Nhap so soi cua dong thu nhat = ');Readln(a[1]);
 Write(' Nhap so soi cua dong thu hai = ');Readln(a[2]);
 Write(' Bandi truoc C/K ');Readln(ch);
 Repeat
  If upcase(ch)='C' then
   Begin
        Writeln(' Dong 1 co: ',a[1],' vien soi');
        Writeln(' Dong 2 co: ',a[2],' vien soi');
        If (a[1]=1) and (a[2]=1) then
          Begin
               Writeln('Ban da thua ');
               Readln;
               exit;
          End;
        Repeat
          Write(' Ban chia dong nao ');Readln(d);
        Until (d>0) and (d<3) and (a[d]>1);
        If d=1 then k:=2 else k:=1;
        repeat
           Write(' Mot trong hai dong do co bao nhieu vien
soi');Readln(c);
        Until c<a[d];
        a[k]:=c;
        a[d]:=a[d]-c;
        Writeln(' Dong 1 co: ',a[1],' vien soi');
        Writeln(' Dong 2 co: ',a[2],' vien soi');
        ch:='K';
   End
  else
     Begin
          If (a[1]=1) and (a[2]=1) then
```

```
            Begin
                Writeln('May da thua');
                Readln;
                exit;
            End;
          d:=a[1];
          c:=1;
          If a[1]<a[2] then
            Begin
                d:=a[2];
                c:=2;
            End;
          If a[1] mod 2=0 then
            Begin
                d:=a[1];
                c:=1;
            End;
          Writeln(' May chia dong ',c);
          a[2]:=1;
          a[1]:=d-1;
          ch:='C';
      End;
  Until false;
END.
```

15. Program P31115;{boc diem}

```
Uses crt;
Var n:byte;
    ch:char;

Function bandi:integer;
Var a:integer;
Begin
  Repeat
    write('Ban boc may que 1-10: ');readln(a);
  Until (a>0) and (a<11) and (a<=n);
  bandi:=a;
End;
Function maydi:integer;
Var a:integer;
Begin
  a:=n mod 11;
```

```
  If a=0 then a:=1;
  maydi:=a;
  writeln('May boc ',a,' que ');
End;

BEGIN
  Clrscr;
  n:=100;
  write(' Ban di truoc C/K ');readln(ch);
  Repeat
   If upcase(ch)='C' then
     Begin
         If n=0 then
           Begin
                write(' Ban da thua');
                readln;
                halt;
           End;
         n:=n-bandi;
         writeln(' Dong soi con ',n,' vien');
         ch:='K';
   End
  else
     Begin
          If n=0 then
             Begin
                  write(' May da thua');
                  readln;
                  halt;
             End;
          n:=n-maydi;
          writeln(' Dong soi con ', n ,' vien');
          ch:='C';
     End;
  Until false;
End.
```

17. Program P31117;{chon thoi gian}

```
Uses crt;
Var n,t,n1,t1:integer;
     ch:char;
Begin
```

```
 Repeat
   Write(' Ban cho ngay ');Readln(n);
 Until (n>0) and (n<32);
 t:=0;
 Write(' Ban choi truoc C/K');Readln(ch);
 Repeat
  if upcase(ch)='C' then
    Begin
         Writeln(' Ngay ',n,' thang ',t);
         if (n=31) and (t=21) then
           Begin
                Writeln(' Ban da thua');
                Readln;
                exit;
           End;
         Repeat
            Write(' Nhap ngay moi ');Readln(n1);
            Write(' Nhap thang moi');Readln(t1);
         Until (n1<32) and (t1<13) and (n1>=n) and (t1>=t) and
((n1<>n) or (t1<>t));
         n:=n1;
         t:=t1;
    End
  else
     Begin
          Writeln(' Ngay ',n,' thang ',t);
          if (n=31) and (t=21) then
             Begin
                  Writeln(' May da thua');
                  Readln;
                  exit;
             End;
     End;
 Until false;
End.
```

19. Program P31119;{Tro choi Nim}

```
Uses crt;
Var  n:byte;
     a:array[1..100] of word;

Procedure input;
```

```
Var i:byte;
Begin
  Writeln;
  Write(' Nhap so dong soi N= ');Readln(n);
  For i:=1 to n do
    begin
          Write(' Nhap so soi cua dong ',i,' = ');
          Readln(a[i]);
    end;
End;
Procedure nguoichoi;
Var c,b:integer;
Begin
  Repeat
    Write(' Ban boc dong nao ');Readln(c);
    If a[c]=0 then Writeln(' Dong do khong con soi');
  Until a[c]<>0;
  Repeat
    Write(' Ban boc may vien ');Readln(b);
    If b>a[c] then Writeln(' Khong the boc nhieu hon so soi ');
  Until b<=a[c];
  a[c]:=a[c]-b;
End;
function over:boolean;
Var i:byte;
Begin
  over:=false;
  For i:=1 to n do If a[i]<>0 then exit;
  over:=true;
End;
Procedure maychoi;
Var i,t:byte;
      d,d1,max:word;
Begin
  max:=0;
  For i:=1 to n do
    If a[i]>max then
      Begin
            max:=a[i];
            t:=i;
      End;
```

```
 d:=0;
 For i:=1 to n do
  If i<>t then d:=d xor a[i];
 For d1:=1 to a[t] do If d1 xor d = 0 then break;
 d1:=a[t]-d1;
 If d1=0 then d1:=1;
 a[t]:=a[t]-d1;
 Writeln(' May boc dong ',t,' voi so soi la ',d1,' vien');
End;
Procedure play;
Var ch:char;
     i:byte;
Begin
 Write('Ban di truoc C/K');Readln(ch);
 Repeat
  If upcase(ch)='C' then
   Begin
        For i:=1 to n do Write(' ',a[i]);
        Writeln;
        If over then
         Begin
              Writeln(' Ban da thua');
              Readln;
              halt;
         End;
        nguoichoi;
        ch:='K';
    End
   else
     Begin
         For i:=1 to n do Write(' ',a[i]);
         Writeln;
         If over then
           Begin
                Writeln(' May da thua');
                Readln;
                halt;
           End;
         maychoi;
         ch:='C';
     End;
  Until false;
```

```
End;

BEGIN
 Clrscr;
 input;
 play;
END.
```

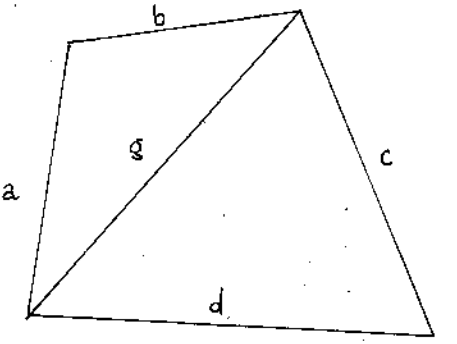
21. Program P31121;{Tro choi Keylesa}

```
Uses crt;
Var n,i:byte;
     ch:char;
     a:array[1..200] of byte;
Function doixung:boolean;
Var i:byte;
Begin
 doixung:=false;
 For i:=1 to n div 2 do If a[i]<>a[n-i+1] then exit;
 doixung:=true;
End;

Procedure hien;
Var i:byte;
Begin
 clrscr;
 For i:=1 to n do
   begin
         gotoxy(i*6,3);
         textcolor(15);
         If a[i]=1 then textcolor(4);
         write('*');
   end;
 writeln;
End;

Function over:boolean;
Var i:byte;
Begin
 over:=false;
 For i:=1 to n do If a[i]=1 then exit;
 over:=true;
End;
```

```
Procedure nguoidi;
Var c,b:byte;
Begin
hien;
Repeat
  write(' Ban boc vien thu may ');readln(c);
  write(' Ban boc may vien ');readln(b);
  b:=b-1;
Until (a[c]=1) and (b<2) and (a[c+b]=1);
a[c]:=0;
a[c+b]:=0;
End;

Procedure maydi;
Var i:byte;
Begin
i:=n div 2;
If (n mod 2=0) and (a[i]=1) and (a[i+1]=1) then
  begin
        a[i]:=0;
        a[i+1]:=0;
        exit;
  end;
If (n mod 2<>0) and (a[i+1]=1) then
  begin
       a[i+1]:=0;
       exit;
  end;
For i:=1 to n do
 If a[i]=1 then
   begin
         a[i]:=0;
         If doixung then exit;
         a[i]:=1;
   end;
For i:=1 to n-1 do
 If (a[i]=1) and (a[i+1]=1) then
   begin
        a[i]:=0;
        a[i+1]:=0;
        If doixung then exit;
```

```
        a[i]:=1;
        a[i+1]:=1;
    end;
For i:=1 to n do If a[i]=1 then
  begin
        a[i]:=0;
        exit;
  end;
End;
BEGIN
 textbackground(0);
 clrscr;
 write(' Nhap so vien soi N= ');readln(n);
 write(' Ban di truoc C/K ');readln(ch);
 For i:=1 to n do a[i]:=1;
 a[n+1]:=0;
 Repeat
   If upcase(ch)='C' then
     begin
          If over then
            begin
                  write(' Ban da thua');
                  readln;
                  exit;
            end;
          nguoidi;
          ch:='0';
     end
   else
      begin
           If over then
             begin
                   write(' May da thua');
                   readln;
                   exit;
             end;
           maydi;
           hien;
           ch:='C';
      end;
  Until false;
 END.
```

Bài 12. Các thuật toán hình học



1. Ký hiệu g là đường chéo chỉ ra trên hình vẽ. Xét các tam giác với các cạnh a, b, g và c, d, g ta có các bất đẳng thức sau:

a + b > g > |a - b| (1)

c + d > g > |c - d| (2)

Điều kiện cần và đủ để a, b, c, d tạo ra các cạnh của tứ giác là tồn tại g thoả mãn điều kiện trên. Từ hai bất đẳng thức trên suy ra:

a + b > |c - d| (3)

c + d > |a - b| (4)

Ta sẽ chứng minh rằng các điều kiện (3), (4) cũng là điều kiện đủ để a, b, c, d lập thành các cạnh của một tứ giác. Thật vậy, các bất đẳng thức (3) và (4) tạo thành hai khoảng trên trục số. Nếu hai khoảng này rời nhau thì ta phải có hoặc |a - b| > a + b hoặc |c - d| > c + d. Điều này không thể xảy ra. Vậy hai khoảng trên giao nhau và do đó tồn tại g thỏa mãn (1) và (2). Đó là đpcm.

Chú ý: Điều kiện (3) và (4) tương đương với hệ các bất đẳng thức sau:

a + b + c > d

a + b + d > c

a + c + d > b

b + c + d > a

```
Program P31201;{Kiem tra tu giac voi do dai cac canh lien nhau}
Uses Crt;
Const     fn = 'TUGIAC.INP';
          fg = 'TUGIAC.OUT';
Var  a ,b ,c ,d : real;
Procedure Nhap;
```

```
Var  f : text;
Begin
Assign( f ,fn ); Reset( f );
Readln( f ,a ,b ,c ,d );
Close( f );
End;

Procedure Out( s : string );
Var  f : text;
Begin
Assign( f ,fg ); Rewrite( f );
Write( f ,s );
Close( f );
End;

BEGIN
ClrScr;
Nhap;
If (a + b + c <= d) or (a + b + d <= c) or (a + c + d <= b) or (b + c + d
<= a) or
     (a <= 0) or (b <= 0) or (c <= 0) or(d <= 0) then out( 'Khong' )
  Else out( 'Co' );
END.
```

3. Program P31203;{Kiem tra xem co tao thanh 5 canh cua ngu giac khong}

```
Uses Crt;
Const  fn = 'NGUGIAC.INP';
       fg = 'NGUGIAC.OUT';
Var a, b, c, d, e :    real;
Procedure Nhap;
Var f: text;
Begin
  Assign( f ,fn ); Reset( f );
  Readln( f ,a ,b ,c ,d );
  Close( f );
End;

Procedure Out( s : string );
Var  f : text;
```

```
Begin
  Assign( f ,fg ); Rewrite( f );
  Write( f ,s );
  Close( f );
End;

BEGIN
  ClrScr;
  Nhap;
  If (a + b + c + e <= d) or (a + b + d + e <= c) or (a + c + d + e <= b) or
(b + c + d + e <= a) or
    (a + b + c + d <= e) or (a <= 0) or (b <= 0) or (c <= 0) or (d <= 0) or
(e <= 0) then out( 'Khong' )
   Else Out( 'Co' );
END.
```

5. Program P31205;

```
(*  Tinh so giao diem cua n duong thang   *)
Uses   Crt;
Const      fn = 'DUONG.INP';
           fg = 'DUONG.OUT';
           max = 100;
           exp = 0.0001;
Var  a ,b ,c    : array[1..max] of integer;
     n : integer;
     sgd : integer;

Procedure Nhap;
Var  f: text;
     i: integer;
Begin
  Assign( f ,fn ); Reset( f );
  Readln( f ,n );
  For i := 1 to n do Readln(f ,a[i] ,b[i] ,c[i] ); { ax + by = c }
  Close(f );
End;

Procedure Chuanbi;
Begin
  sgd :=   0;
```

```
End;

Function Giaodiem( i ,j : integer;Var x ,y : real ) : boolean;
Var   d ,dx ,dy : integer;
Begin
 d := a[i] * b[j] - a[j] * b[i];
 dx :=  c[i] * b[j] - c[j] * b[i];
 dy :=  a[i] * c[j] - a[j] * c[i];
 If d <> 0 then  begin x :=  dx/d; y := dy/d; end;
 giaodiem := d <> 0;
End;

Function Giatri( i : integer;x ,y : real ) : real;
Begin
 Giatri := a[i] * x + b[i] * y - c[i];
End;

Function bang( a ,b : real ) : boolean;
Begin
 bang := abs( a - b ) <= exp;
End;

Function Thoaman( i ,j : integer;x ,y : real ) : boolean;
Var  ii: integer;
Begin
 Thoaman := false;
 For ii := 1 to i - 1 do
   If (ii <> j) and bang(giatri(ii ,x ,y),0) then exit;
 Thoaman := true;
End;

Function Catrieng( i : integer ) : integer;
Var    ii , gt:integer;
       x, y : real;
Begin
gt :=  0;
For ii := 1 to i do
  If giaodiem(i ,ii ,x ,y) then
    If thoaman(i ,ii ,x ,y) then Inc(gt);
```

```
catrieng := gt;
End;

Procedure Tinhsl;
Var    i : integer;
Begin
 For i := 1 to n do Inc(sgd ,catrieng(i));
End;

Procedure Ghinhan;
Var   f: text:
Begin
 Assign(f ,fg); Rewrite(f);
 Write(f ,sgd);
 Close(f);
End;

BEGIN
 ClrScr;
 Nhap;
 Chuanbi;
 Tinhsl;
 ghinhan;
END.
```

7. Program P31207;{Tinh va ve ra tat ca Mino}

```
Uses Crt;
Const   fn = 'NMINO.INP';
        fg = 'NMINO.OUT';
        max = 16;
Type    bang =   array[0..max+1,0..max+1] of integer;
Var   n : integer;
      lonmin    : integer;
      hinh ,hinh1 ,xet ,dd :   bang;
      hang ,cot: array[1..max] of integer;
      sl : integer;
      qi,qj :   array[1..max*max] of integer;
      sh ,sc   :integer;
      hangthieu , cotthieu:integer;
      slch :   longint;
```

```
    f : text;

Procedure Nhap;
Var  f:text;
Begin
 Assign(f,fn); Reset(f);
 Readln(f ,n);
 Close(f);
End;

Procedure Chuanbi;
Begin
 lonmin:= trunc(sqrt(n));
 If n <> sqr(lonmin) then Inc(lonmin);
 slch := 0;
End;

Function min2( a ,b : integer ) : integer;
Begin
 If a < b then min2 := a Else min2 := b;
End;

Procedure Taobien( i ,j : integer );
Var  ii ,jj : integer;
Begin
 FillChar(dd ,SizeOf(dd),1);
 FillChar(xet,SizeOf(xet),1);
 For ii := 1 to i do
  For jj := 1 to j do
   begin
        dd[ii,jj] := 0;
        xet[ii,jj]   :=   0;
   end;
End;

Procedure Ghinhancauhinh;
Var  i ,j :   integer;
Begin
 Inc(slch);
 Writeln(f,sh ,' ' ,sc);
```

```
 For i := 1 to sh do
  begin
      For j := 1 to sc do Write(f,(dd[i,j] mod 2):2);
      Writeln(f)
  end;
End;

Procedure Quaytrai;
Var  hinh1 : bang;
     i,j : integer;
Begin
 hinh1 := hinh;
 For i := 1 to sh do
   For j := 1 to sc do hinh[i,j] := hinh1[sc-j+1,i];
End;

Procedure Lathinh;
Var  hinh1 : bang;
     i ,j : integer;
Begin
 hinh1:= hinh;
 For i := 1 to sh do
   For j := 1 to sc do hinh[i,j] := hinh1[sh-i+1,sc-j+1];
End;

Procedure Daohinh;
Var  hinh1 : bang;
     i,j : integer;
Begin
 hinh1 := hinh;
 For i := 1 to sh do
   For j := 1 to sc do hinh[i,j] := hinh1[sh-i+1,j];
End;

Function Bethat : boolean;
Var  ii,jj   :integer;
Begin
 Bethat := false;
 For ii := 1 to sh do
   For jj := 1 to sc do
```

```
    If hinh[ii,jj] <> hinh1[ii,jj] then
        begin
            Bethat:= hinh[ii,jj] < hinh1[ii,jj];
            exit;
        end;
End;

Function Behon : boolean;
Begin
 Behon := Bethat;
End;

Function Xethinhvuong : boolean;
Begin
 Xethinhvuong :=    false;
 Quaytrai;
 If Behon then exit; Quaytrai;
 If Behon then exit; Quaytrai;
 If Behon then exit; Daohinh;
 If Behon then exit; Quaytrai;
 If Behon then exit; Quaytrai;
 If Behon then exit; Quaytrai;
 If Behon then exit; Xethinhvuong := true;
End;

Function Xetchunhat : boolean;
Begin
 Xetchunhat := false;
 Lathinh;
 If Behon then exit; Daohinh;
 If Behon then exit; Lathinh;
 If Behon then exit; Xetchunhat := true;
End;

Procedure Chuyensang( a : bang;Var b : bang );
Var       i,j:integer;
Begin
For i := 1 to sh do
 For j := 1 to sc do b[i,j] := a[i,j] mod 2;
```

```
End;

Procedure Thughinhancauhinh;
Begin
  Chuyensang(dd ,hinh);
  hinh1:=   hinh;
  If sh = sc then   begin   If not Xethinhvuong then exit;  end
     Else If not Xetchunhat then exit;
  Ghinhancauhinh;
End;

Procedure Xetthem( i ,j : integer );
Begin
  Inc(xet[i,j]);
  If xet[i,j] = 1 then
     begin
          Inc(sl);
          qi[sl] := i;
          qj[sl] := j
     end;
End;

Procedure Xetbot( i ,j : integer );
Begin
  If xet[i,j] = 1 then Dec(sl);
  Dec( xet[i,j] );
End;

Procedure Themdiem( ii : integer );
Var  i ,j : integer;
Begin
  i := qi[ii];
  j := qj[ii];
  dd[i,j] := 1;
  If dd[i,j-1] = 0 then Xetthem(i ,j-1);
  If dd[i,j+1] = 0 then Xetthem(i ,j+1);
  If dd[i-1,j] = 0 then Xetthem(i-1,j);
  If dd[i+1,j] = 0 then Xetthem(i+1,j);
End;
```

```
Procedure Bodiem( ii : integer );
Var  i , j    : integer;
Begin
  i := qi[ii];
  j := qj[ii];
  dd[i,j] := 0;
  If dd[i,j-1] = 0 then Xetbot(i,j-1);
  If dd[i,j+1] = 0 then Xetbot(i,j+1);
  If dd[i-1,j] = 0 then Xetbot(i-1,j);
  If dd[i+1,j] = 0 then Xetbot(i+1,j);
End;

Procedure Xethangcot( ii : integer );
Var  i ,j    :integer;
Begin
  i := qi[ii];
  j := qj[ii];
  Inc(hang[i]);
  If hang[i] = 1 then Dec(hangthieu);
  Inc(cot[j]);
  If cot[j] = 1 then Dec(cotthieu);
End;

Procedure Xetlaihangcot( ii : integer );
Var  i,j :    integer;
Begin
  i := qi[ii];
  j := qj[ii];
  If hang[i] = 1 then Inc(hangthieu);
  Dec(hang[i]);
  If cot[j] = 1 then Inc(cotthieu);
  Dec(cot[j]);
End;
Procedure Duyet( i : integer;last : integer );
Var  ii :integer;
Begin
If i > n then
   begin  thughinhancauhinh;  exit; end;
For ii := last + 1 to sl do
   begin
        themdiem(ii);
```

```
        xethangcot(ii);
        If hangthieu + cotthieu <= n - i then duyet(i+1,ii);
        Xetlaihangcot(ii);
        bodiem(ii);
  end;
End;

Procedure Duyetcauhinh( i ,j : integer );
Var jj : integer;
Begin
 sh := i;
 sc := j;
 FillChar(hang ,SizeOf(hang),0);
 FillChar(cot,SizeOf(cot),0);
 hangthieu := sh;
 cotthieu := sc;
 taobien(i ,j);
 For jj := 1 to j do
   begin
        sl:= 1;
        qi[1] := 1;
        qj[1] := jj;
        duyet(1,0);
        dd[1,jj] := 2;
   end;
End;

Procedure Duyethinhbao;
Var i ,j : integer;
    minj ,maxj : integer;
Begin
For i := lonmin to n do
  begin
       minj := (n-1) div i + 1;
       maxj := min2(n+1-i,i);
       For j := minj to maxj do duyetcauhinh(i,j);
  end;
End;

Procedure Ghicuoi;
Var f : file of char;
```

```
    s : string;
    i : integer;
Begin
 str(slch,s);
 Assign(f,fg); reset(f);
 Seek(f,0);
 For i := 1 to length(s) do  Write(f,s[i]);
 Close(f);
End;

BEGIN
 Clrscr;
 Assign(f,fg); Rewrite(f);
 Writeln(f ,' ');
 Nhap;
 Chuanbi;
 duyethinhbao;
 Close(f);
 ghicuoi;
END.
```

9. Program P31209;

```
Uses Crt;
Const     fn = 'THUOCTHO.INP';
          fg = 'THUOCTHO.OUT';
          max = 100;
          chuan : array[1..3] of integer = ( 1 ,1 ,1 );
Var    m,n:integer;
       a:array[1..max,1..max] of integer;
       sl:integer;

Procedure Nhap;
Var  f: text;
Begin
 Assign(f ,fn); Reset(f);
 Readln(f,m,n);
 Close(f);
End;

Procedure Chuanbi;
Begin
```

```
 FillChar(a ,SizeOf(a),0);
 sl:= 0;
End;

Procedure Pathinhngang( i ,j ,c : integer );
Var   ii:integer;
Begin
 For ii := 1 to 3 do  a[i,j+ii-1] := chuan[ii] + c;
End;

Procedure Dathinhdoc( i ,j ,c : integer ):
Var   ii:    integer;
Begin
 For ii := 1 to 3 do  a[i+ii-1,j] := chuan[ii] + c;
End;

Procedure Datngang( i1 ,j1 ,i2 ,j2 : integer );
Var  i,j :    integer;
Begin
For i := i1 to i2 do
 For j := j1 to j2 do
    If a[i,j] = 0 then
       begin
            Pathinhngang(i,j,sl);
            Inc(sl);
       end;
End;

Procedure Datdoc( i1 ,j1 ,i2 ,j2 : integer );
Var  i,j : integer;
Begin
For i := i1 to i2 do
 For j := j1 to j2 do
    If a[i,j] = 0 then
       begin
            dathinhdoc(i,j,sl);
            Inc(sl );
       end;
End;
```

```
Procedure Datvaohcn( i1 ,j1 ,i2 ,j2 : integer );
Begin
If (i2-i1+1) mod 3 = 0 then datdoc(i1,j1,i2,j2)
 Else datngang(i1,j1,i2,j2);
End;

Procedure Ghinhan;
Var  f:text;
     i,j:integer;
Begin
 Assign(f,fg); Rewrite(f);
 For i := 1 to n do
  begin
       For j := 1 to n do Write(f,a[i,j]:6);
       Writeln(f);
  end;
 Close(f);
End;

BEGIN
 Clrscr;
 Nhap;
 Chuanbi;
 Datvaohcn(1,1,m,n);
 Ghinhan
END.
```

11. Program P31211; {Xep hinh vuong}

```
Uses Crt;
Const   fn = 'XEPVUONG.INP';
        fg = 'XEPVUONG.OUT';
        max = 10;
        maxc =  100;
Type    cap = record
              i,j:integer;
            end;
Var   n ,k :  integer;
      a : array[1..max+1] of integer;
      ddmax ,dd : array[0..maxc+1,0..maxc+1] of byte;
      moc :  array[0..maxc+1,0..maxc+1] of byte;
```

```
        tong :  integer;
        thuamax : integer;
       chon :  array[1..max+1] of byte;
       cohinh : boolean;
       slduoc : integer;
       tapmoc : array[1..maxc] of cap;
       slmoc : integer;

Procedure Nhap;
Var  f:text;
     i :integer;
Begin
  Assign(f,fn); Reset(f);
  Readln(f,n,k);
  For i := 1 to n do Read(f,a[i]);
  Close(f);
End;

Procedure Chuanbi;
Var  i,j :  integer;
Begin
  FillChar(dd ,SizeOf(dd),1);
  FillChar(chon,SizeOf(chon),0);
  FillChar(moc,SizeOf(moc),0);
  For i := 1 to k do
    For j := 1 to k do  dd[i,j] := 0;
  tong := 0;
  For i := 1 to n do  Inc(tong ,a[i]*a[i]);
  thuamax:=trunc(sqrt(k*k-tong));
  moc[1,1] := 1;
  slduoc:=0;
  slmoc:=1;
  tapmoc[1].i:= 1;
  tapmoc[1].j:= 1;
End;

Function vua( k ,i ,j : integer ) : boolean;
Var  ii, jj : integer;
Begin
vua :=  false;
For ii := 1 to a[k] do
```

```
  For jj := 1 to a[k] do If dd[i+ii-1,j+jj-1] <> 0 then exit;
vua:= true;
End;

Procedure Datmocngangduoi( k ,i ,j : integer );
Var  ii, jj  : integer;
Begin
 jj :=j + a[k] - 1;
 ii :=  i + a[k];
 If dd[ii,jj] = 0 then
   begin
        While dd[ii,jj-1] = 0 do dec(jj);
         Inc(moc[ii,jj]);
         If moc[ii,jj] = 1 then
           begin
                 Inc(slmoc);
                 tapmoc[slmoc].i    := ii;
                 tapmoc[slmoc].j := jj;
           end;
   end;
End;

Procedure Datmocdocphai( k ,i ,j : integer );
Var  ii,jj:integer;
Begin
 jj :=j + a[k];
 ii :=i + a[k] - 1;
 If dd[ii,jj] = 0 then
    begin
         While dd[ii-1,jj] = 0 do Dec(ii);
          Inc(moc[ii,jj]);
          If moc[ii,jj] = 1 then
            begin
                  Inc( slmoc );
                   tapmoc[slmoc].i  := ii;
                   tapmoc[slmoc].j  := jj;
            end;
   end;
End;
```

```
Procedure Bomocngangduoi( k ,i ,j : integer );
Var  ii,jj    : integer;
Begin
 jj := j + a[k] - 1;
 ii := i + a[k];
 If dd[ii,jj] = 0 then
   begin
         While dd[ii,jj-1] = 0 do dec(jj);
         If moc[ii,jj] = 1 then Dec(slmoc);
         Dec( moc[ii,jj] );
    end;
End;

Procedure bomocdocphai( k ,i ,j : integer );
Var  ii,jj    :integer;
Begin
 jj :=  j + a[k];
 ii :=  i + a[k] - 1;
 If dd[ii,jj] = 0 then
   begin
       While dd[ii-1,jj] = 0 do Dec(ii);
       If moc[ii,jj] = 1 then Dec(slmoc);
       Dec(moc[ii,jj]);
  end;
End;
Procedure Dat( k ,i ,j : integer );
Var  ii, jj : integer;
Begin
For ii := 1 to a[k] do
  For jj := 1 to a[k] do dd[i+ii-1,j+jj-1] := k;
datmocngangduoi(k,i ,j);
datmocdocphai(k,i,j);
End;

Procedure Bo( k ,i ,j : integer );
Var ii, jj : integer;
Begin
  Bomocngangduoi(k ,i ,j);
  Bomocdocphai(k ,i ,j);
  For ii := 1 to a[k] do
    For jj := 1 to a[k] do dd[i+ii-1,j+jj-1] := 0;
```

```
End;

Procedure Duyet( last : integer ); forward;
Procedure Datthuhinh( k ,i ,j : integer );
Var  ii: integer;
Begin
For ii := 1 to n do
  If chon[ii] = 0 then
    If vua(ii ,i ,j) then
      begin
            dat(ii,i,j);
            chon[ii] := 1;
            Inc(slduoc);
            duyet(k);
            Dec(slduoc);
            chon[ii] := 0;
            bo(ii,i,j);
            If cohinh then exit;
      end;
End;
Procedure Ghinhanhinhmoi;
Begin
 ddmax := dd;
End;
Procedure Duyet( last : integer );
Var  i:integer;
Begin
If slduoc >= n then
  begin
        Ghinhanhinhmoi;
        cohinh := true;
        exit;
   end;
For i := last+1 to slmoc do Datthuhinh(i,tapmoc[i].i ,tapmoc[i].j);
End;

Function Thoamancanh( m : integer ) : boolean;
Begin
 cohinh := false;
 Inc(n);
 a[n] := m;
```

```
  duyet(0);
  Dec(n);
  Thoamancanh := cohinh;
End;

Procedure Duyetnhiphan;
Var  d,c: integer;
     m : integer;
Begin
d:= 0;
c:= thuamax;
While d <= c do
  begin
      m := (d + c) div 2;
      If Thoamancanh(m) then
        begin
             Writeln('Best now : ',m );
             d := m + 1;
        end
       Else c := m - 1;
  end;
End;

Procedure Ghinhan;
Var  i,j:   integer;
     f: text;
Begin
Assign(f ,fg); Rewrite(f);
For i := 1 to k do
 begin
     For j := 1 to k do Write(f ,ddmax[i,j] mod (n+1):4);
     Writeln(f);
 end;
Close(f);
End;

BEGIN
 ClrScr;
 Nhap;
 Chuanbi;
 Duyetnhiphan;
```

```
 Ghinhan;
END.
```

13. Program P31213;

```
{Tim phu cua mot hinh chu nhat boi cac HCN khong so sanh
duoc}
Uses  Crt;
Const fn = 'CHUNHAT.INP';
      fg = 'CHUNHAT.OUT';
      max = 100;
Type  chunhat = record
              i,j: integer;
            end;
Var   m,n : integer;
      dt  : integer;
      sh  : integer;
      hinh: array [1..max] of chunhat;
      slch: integer;
      chon: array [1..max] of byte;
      dd  : array [0..max+1,0..max+1] of byte;

Procedure Nhap;
Var f: text;
Begin
  Assign(f,fn); Reset(f);
  Readln(f,m,n);
  Close(f);
End;

Procedure Chuanbi;
Var i,j: integer;
Begin
 dt:=m*n;
 slch:=0;
 FillChar(chon,SizeOf(chon),0);
 FillChar(dd,SizeOf(dd),1);
 For i:=1 to m do
   For j:=1 to n do dd[i,j]:=0;
End;

Procedure Ghinhanhinh;
Var i,j: integer; f: text;
```

```
Begin
 Assign(f,fg); Reset(f);
 For i:=1 to m do
 begin
     For j:=1 to n do Write(f,dd[i,j]:3);
     exit;
 end;
 Close(f);
End;

Function vua(k,i,j: integer): boolean;
Var ii,jj: integer;
Begin
 vua:=false;
 For ii:=1 to hinh[k].i do
   For jj:=1 to hinh[k].j do
     If dd[i+ii-1,j+jj-1]<>0 then exit;
 vua:=true;
End;

Procedure dat(k,i,j: integer);
Var ii,jj: integer;
Begin
 Chon[k]:=1;
 For ii:=1 to hinh[k].i do
   For jj:=1 to hinh[k].j do dd[i+ii-1,j+jj-1]:=k;
End;

Procedure bo(k,i,j: integer);
Var ii,jj: integer;
Begin
 For ii:=1 to hinh[k].i do
   For jj:=1 to hinh[k].j do dd[i+ii-1,j+jj-1]:=0;
 Chon[k]:=0;
End;

Procedure Timdiemtiep(Var ii,jj: integer);
Begin
 While dd[ii,jj]<>0 do
   begin
         Inc(jj);
         If jj>n then begin Inc(ii); jj:=1; end;
```

```
  end;
End;

Procedure Duyetcachdat(i: integer; ii,jj: integer);
Var k: integer;
Begin
 If i>sh then
  begin
        ghinhanhinh;
        halt;
  end;
 timdiemtiep(ii,jj);
 For k:=1 to sh do
  If chon[k]=0 then
   If vua(k,ii,jj) then
    begin
         dat(k,ii,jj);
         duyetcachdat(i+1,ii,jj);
         bo(k,ii,jj);
    end;
End;

Procedure Chonhinh(i,j: integer);
Begin
 Inc(sh);
 hinh[sh].i:=i; hinh[sh].j:=j;
 Dec(dt,i*j);
End;

Procedure Bohinh(i,j: integer);
Begin
 Inc(dt,i*j);
 Dec(sh);
End;

Function Lonhon(i,j,dt: integer): boolean;
Var ii,jj,tong: integer;
Begin
 ii:=m; jj:=j-1;
 tong:= 0;
 While (ii>i)and(jj>0)and(tong<dt) do
  begin
```

```
        Inc(tong,ii*jj);
        Dec(ii); Inc(jj);
  end;
 lonhon:=tong>=dt
End;

Procedure Duyettong(i: integer; lasti,lastj: integer);
Var ii,jj: integer;
Begin
 If dt=0 then
  begin
        Inc(slch);
        duyetcachdat(1,1,1);
        exit;
  end;
 For ii:=lasti+1 to m do
  For jj:=lastj-1 downto 1 do
   If (dt>=ii*jj)and(ii+jj<m+n) then
     begin
        chonhinh(ii,jj);
        If lonhon( ii,jj,dt) then duyettong(i+1,ii,jj);
        bohinh(ii,jj);
     end;
End;

Procedure Vonghiem;
Var f: text;
Begin
 Assign(f,fg); Rewrite(f);
 Write(f,'VO NGHIEM');
 Close(f);
End;

BEGIN
 ClrScr;
 Nhap;
 Chuanbi;
 Duyettong(1,0,n+1);
 Vonghiem;
END.
```

15. Program P31215;{Kiem tra diem M nam trong hay ngoai da giac}

```
Uses Crt;
Const  fn = 'DAGIAC.INP';
       fg = 'DAGIAC.OUT';
       max = 100;
Var   x,y: array[0..max+1] of real;
      n:integer;
      a,b,c:array[1..max] of real;
      xd ,yd    :real;

Procedure Nhap;
Var  f:text;
     i:integer;
Begin
  Assign(f,fn);Reset(f);
  Readln(f,n);
  For i := 1 to n do Readln(f,x[i],y[i]);
  Readln(f,xd,yd);
  Close(f);
End;

Procedure Chuanbi;
Begin
  x[0] := x[n];
  y[0] := y[n];
  x[n+1] := x[1];
  y[n+1] := y[1];
End;

Procedure Lapft( i : integer );
Begin
  a[i] := y[i-1] - y[i];
  b[i] := x[i] - x[i-1];
  c[i] := -(a[i]*x[i] + b[i]*y[i]);
End;

Procedure Tinhft;
Var i : integer;
Begin
  For i := 1 to n do lapft(i);
```

```
End;

Function ft( k : integer;x ,y : real ) : real;
Begin
  ft:= a[k]*x + b[k]*y + c[k];
End;

Function namtren : boolean;
Var  i:integer;
Begin
  namtren := true;
  For i := 1 to n do
    If  (ft(i,xd,yd)=0) and ((xd - x[i-1])*(xd - x[i])<=0) and ((yd - y[i-
1])*(yd - y[i])<=0) then exit;
  namtren := false;
End;
Function catthang( i : integer ) : boolean;
Begin
  Catthang := false;
  If (x[i-1] >= xd) or (x[i] < xd) then exit;
  If (x[i] = xd) and ((x[i-1]- xd)*(x[i+1]-xd) >= 0) then exit;
  catthang := true;
End;
Function namtrong : boolean;
Var i:integer;
      tong : integer;
Begin
  tong := 0;
  For i := 1 to n do
    If catthang(i) then Inc(tong);
  namtrong := tong mod 2 = 1;
End;

Procedure ra( s : string );
Var  f:text;
Begin
  Assign(f,fg);Rewrite(f);
  Write(f,s);
  Close(f);
End;
```

```
BEGIN
 Clrscr;
 Nhap;
 Chuanbi;
 Tinhft;
 If not namtren then
   begin
        If namtrong then  ra( 'Diem M nam trong da giac.' )
          Else  ra( 'Diem M nam ngoai da giac.' )
   end
  Else ra( 'Diem M nam tren canh da giac.' );
END.
```

17. Program P31217;{Chi ra cac hinh chu nhat long nhau}

```
Uses Crt;
Const  fn = 'LONGNHAU.INP';
       fg = 'LONGNHAU.OUT';
       max =100;
Var  x1 ,y1 ,x2 ,y2 : array[1..max] of integer;
     n :  integer;
     dt :  array[1..max] of longint;
     cs :  array[1..max] of integer;
     con ,cha ,chung : array[1..max] of integer;
     longmax : integer;
Procedure Nhap;
Var  f: text;
     i: integer;
Begin
 Assign(f,fn); Reset(f);
 Readln(f,n);
 For i := 1 to n do Readln( f ,x1[i] ,y1[i] ,x2[i] ,y2[i] );
     (* Dieu kien : x2[i] >= x1[i] ,y2[i] >= y1[i] ,i = 1..n *)
 Close(f);
End;
Procedure Chuanbi;
Var i: integer;
Begin
For i := 1 to n do
  begin
       cs[i] := i;
       dt[i] := x2[i] - x1[i];
       dt[i] := dt[i] * (y2[i] - y1[i]);
```

```
  end;
End;
Procedure Doil( Var a ,b : longint );
Var  h : longint;
Begin
 h := a;
 a := b;
 b := h;
End;

Procedure Doii( Var a ,b : integer );
Var  h: integer;
Begin
 h := a;
 a := b;
 b := h;
End;

Procedure Sort;
Var  i ,j:   integer;
Begin
For i := 1 to n - 1 do
 For j := i + 1 to n do
   If dt[i] > dt[j] then
     begin
          Doil(dt[i],dt[j]);
          Doii(cs[i],cs[j]);
          Doii(x1[i],x1[j]);
          Doii(y1[i],y1[j]);
          Doii(x2[i],x2[j]);
          Doii(y2[i],y2[j]);
     end;
End;
Function Trong( i ,j : integer ) : boolean;
Begin
 Trong := (x1[i] >= x1[j]) and (y1[i] >= y1[j]) and (x2[i] <= x2[j]) and
(y2[i] <= y2[j]);
End;
Function Longconmax( i : integer ) : integer;
Var  ii,max:integer;
Begin
```

```
max :=   0;
For ii := 1 to i - 1 do
  If trong(ii,i) and (max < con[ii]) then max := con[ii];
longconmax := max;
End;

Procedure Tinhlongcon;
Var  i :integer;
Begin
 For i := 1 to n do con[i] := longconmax( i ) + 1;
End;

Function Longchamax( i : integer ) : integer;
Var  ii,max :integer;
Begin
max :=   0;
For ii := i + 1 to n do
  If trong( i ,ii ) and (max < cha[ii]) then max := cha[ii];
longchamax   := max;
End;

Procedure Tinhlongcha;
Var  i : integer;
Begin
 For i := n downto 1 do  cha[i] := longchamax( i ) + 1;
End;
Procedure Tinhlongchung;
Var  i : integer;
Begin
 For i := 1 to n do  chung[i] := con[i] + cha[i] - 1;
End;
Procedure Tinhmax;
Var  i : integer;
Begin
longmax := 0;
For i := 1 to n do
 If longmax < chung[i] then longmax := chung[i];
End;
Function Chiso( i : integer ) : integer;
Var  ii: integer;
Begin
```

```
For ii := 1 to n do
  If cs[ii] = i then
    begin
        chiso := ii;
        exit;
    end;
End;

Procedure Ghinhan;
Var  f : text;
     i : integer;
Begin
  Assign(f,fg); Rewrite(f);
  Writeln(f ,longmax);
  For i := 1 to n do Write(f,chung[chiso(i)] ,' ' );
  Close(f);
End;

BEGIN
  ClrScr;
  Nhap;
  Chuanbi;
  Sort;
  Tinhlongcon;
  Tinhlongcha;
  Tinhlongchung;
  Tinhmax;
  Ghinhan;
END.
```

Bài 13. Các thuật toán sắp xếp

1. Program P31301; {Sap xep cac qua bong}

```
Uses  Crt;
Const  Nmax=1000;
       inp='P31301.inp';
Var  f: text;
     n,sx,sd,sv,i: integer;
     a: array [1..nmax] of char;

Procedure Nhap;
Var i: integer;
Begin
  sx:=0; sd:=0; sv:=0;
  Assign(f,inp); Reset(f);
  Readln(f,n);
  For i:=1 to n do
   begin
       Readln(f,a[i]);
       Case a[i] of
           'X': inc(sx);
           'D': inc(sd);
           'V': inc(sv);
         End;
   end;
  Close(f);
End;

BEGIN
  Clrscr;
  Nhap;
  for i:=1 to sx do  if (a[i]='D') or (a[i]='V') then a[i]:='X';
  for i:= sx+1 to sd+sx do if (a[i]='X') or (a[i]='V') then a[i]:='D';
  for i:=sd+sx+1 to n do  if (a[i]='D') or (a[i]='X') then a[i]:='V';
  for i:=1 to n do Write(a[i]);
 Readln;
END.
```

3. Program P31303; {Sap xep noi bot}

```
Uses Crt;
Const   Inp='P31303.Inp';
        Out='P31303.Out';
```

```
        Nmax=1000;
Var F: text;
    N: integer;
    A: array [1..Nmax] of integer;

Procedure Nhap;
Var i: integer;
Begin
 Assign(F,Inp); Reset(F);
 Readln(F,N);
 For i:=1 to N do Read(F,A[i]);
 Close(F);
End;

Procedure SapXep;
Var i,j,t: integer;
Begin
 For i:=1 to N-1 do
   For j:=1 to N-1 do
    If A[j]>A[j+1] then
     begin
         t:=A[j];
         A[j]:=A[j+1];
         A[j+1]:=t;
     end;
End;

Procedure InKq;
Var i: integer;
Begin
 Assign(F,Out); Rewrite(F);
 For i:=1 to N do Writeln(F,A[i]);
 Close(f);
End;

BEGIN
  Nhap;
 SapXep;
 InKq;
END.
```

```
4. Program P31304; {Sap xep chon}
Uses Crt;
Const   Inp='P31303.Inp';
        Out='P31304.Out';
        Nmax=1000;
Var F: text;
    N: integer;
    A: array [1..Nmax] of integer;

Procedure Nhap;
Var i: integer;
Begin
 Assign(F,Inp); Reset(F);
 Readln(F,N);
 For i:=1 to N do Readln(F,A[i]);
 Close(F);
End;

Procedure SapXep;
Var i,j,k,t: integer;
Begin
 For i:=1 to N-1 do
  begin
       k:=i;
       For j:=i+1 to N do If A[j]<A[k] then k:=j;
       If k<>i then
          begin
             t:=A[i];
             A[i]:=A[k];
             A[k]:=t;
          end;
  end;
End;

Procedure InKq;
Var i: integer;
Begin
 Assign(F,Out); Rewrite(F);
 For i:=1 to N do Writeln(F,A[i]);
 Close(f);
End;
```

```
BEGIN
 Nhap;
 SapXep;
 InKq;
END.
```

5. Program P31305; {Sap xep chen don}

```
Uses Crt;
Const  Inp='P31303.Inp':
       Out='P31305.Out':
       Nmax=1000;
Var F: text;
    N: integer;
    A: array [0..Nmax] of integer;

Procedure Nhap;
Var i: integer;
Begin
 Assign(F,Inp); Reset(F);
 Readln(F,N);
 For i:=1 to N do Readln(F,A[i]);
 Close(F);
End;

Procedure SapXep;
Var i,j,t: integer;
Begin
 For i:=2 to N do
  begin
       t:=A[i];
       A[0]:=A[i];
       j:=i-1;
       While t<A[j] do
         begin
              A[j+1]:=A[j];
              j:=j-1;
         end;
       A[j+1]:=t;
   end;
End;

Procedure InKq;
```

```
Var i: integer;
Begin
  Assign(F,Out); Rewrite(F);
  For i:=1 to N do Writeln(F,A[i]);
  Close(f);
End;
BEGIN
  Nhap;
  SapXep;
  InKq;
END.
```

7. Program P31307; {chen nhi phan}

```
Uses crt;
Const   fi = 'P31307.inp';
        fo = 'P31307.out';
Var a,b: array [0..10000] of integer;
    n,d: integer;
    f: text;
Procedure chen(k,t: integer);
Var i: integer;
Begin
  Inc(d);
  For i:=d downto t+1 do b[i]:=b[i-1];
  b[t]:=k;
End;

Procedure input;
Var i: integer;
Begin
  Assign(f,fi); Reset(f);
  Readln(f,n);
  For i:=1 to n do Read(f,a[i]);
  Close(f);
End;
Function tim(k: integer): integer;
Var x,y: integer;
Begin
  x:=0;
  y:=d;
  Repeat
    If k>b[(y+x) div 2] then x:=(y+x) div 2
```

```
      Else y:=(y+x) div 2;
  Until (y-x)<2;
  tim:=y;
End;

Procedure work;
Var i: integer;
Begin
  b[0]:=-maxint;
  b[1]:=maxint;
  d:=1;
  For i:=1 to n do chen(a[i],tim(a[i]));
End;

Procedure output;
Var i: integer;
Begin
  Assign(f,fo); Rewrite(f);
  Writeln(f,n);
  For i:=1 to n do Write(f,' ',b[i]);
  Close(f);
End;

BEGIN
  input;
  work;
  output;
END.
```

9. Program P31309; {Thuat toan ghep noi}

```
Uses crt;
Const   Inp ='P31309.inp';
        Out ='P31309.out';
        MNmax=1000;
Var f: text;
    n,m: integer;
    c,d: array [1..MNmax+1] of integer;
    e : array [1..2*MNmax] of integer;

Procedure Nhap;
Var i: integer;
Begin
```

```
 Assign(f,Inp); Reset(f);
 Readln(f,n,m);
 For i:=1 to n do Read(f,c[i]);
 For i:=1 to m do Read(f,d[i]);
 Close(f);
End;

Procedure GhepMang;
Var i,j,k: integer;
Begin
 c[n+1]:=maxint; d[m+1]:=maxint;
 i:=1; j:=1;
 For k:=1 to m+n do
  If c[i]<=d[j] then
      begin
          e[k]:=c[i];
          Inc(i);
      end
    Else
     begin
          e[k]:=d[j];
          inc(j);
      end;
End;

Procedure InKq;
Var i: integer;
Begin
  Assign(f,Out); Rewrite(f);
  Writeln(f,m+n);
  For i:=1 to m+n do Write(f,e[i], ' ');
  Close(f);
End;

BEGIN
  Nhap;
  GhepMang;
  InKq;
END.
```

11. Program P31311; {Tron file}

```
Uses Crt;
```

```
Const  fi1='P31311a.inp';
       fi2='P31311b.inp';
       fo ='P31311.out';
Type   NumberFile = record
                 · x: real;
               end;
Var f,g,q: file of NumberFile;

Procedure mix;
Var x1,x2: numberfile; t1,t2: boolean;
Begin
 t1:= true; t2:= true;
 While (not Eof(f)) or (not Eof(g)) do
   begin
       If Eof(f) and t1 then x1.x:=-100000
         Else If t1 then Read(f,x1);
       If Eof(g) and t2 then x2.x:=-100000
         Else If t2 then Read(g,x2);
       If x1.x<x2.x then
           begin
                t1:=true;
                t2:=false;
                Write(q,x1);
           end
         Else
           begin
                t1:=false;
                t2:=true;
                Write(q,x2);
           end;
   end;
End;
BEGIN
 Assign(f,fi1); Reset(f);
 Assign(g,fi2); Reset(g);
 Assign(q,fo); Rewrite(q);
 mix;
 Close(q); Close(f); Close(g);
END.
```

13. Program P31313; {Sap xep nhanh: Khong de qui}
Uses Crt;

```
Const   Out = 'P31313.out';
        n = 10000;
Type  st = record
          Left,Right:integer;
        end;
Var  f: text;
     a: array [1..n] of integer;

Procedure NgauNhien;
Var i: integer;
Begin
 Randomize;
 For i:=1 to n do a[i]:=Random(maxint);
End;
Procedure Sort;
Var   s,L,R,i,j,x,t : integer;
      stack: array [1..n] of st;
Begin
 s:=1;
 stack[1].Left:=1;
 stack[1].Right:=n;
 Repeat
    L:=stack[s].Left;
    R:=stack[s].Right;
    s:=s-1;
    i:=L; j:=R;
    X:=a[(L+R) div 2];
    Repeat
        While a[i]<X do i:=i+1;
        While a[j]>X do j:=j-1;
        If i<=j then
          begin
                t:=a[i];
                a[i]:=a[j];
                a[j]:=t;
                i:=i+1; j:=j-1;
          end;
    Until i>j;
    If l<j then
      begin
          s:=s+1;
```

```
            stack[s].Left:=l;
            stack[s].Right:=j;
        end;
      If i<r then
        begin
            s:=s+1;
            stack[s].Left:=i;
            stack[s].Right:=r;
        end;
  Until s=0
End;
Procedure InKq;
Var i: integer;
Begin
  Assign(f,out); Rewrite(f);
  For i:=1 to n do Writeln(f,a[i]);
  Close(f);
End;

BEGIN
  NgauNhien;
  Sort;
  InKq;
END.
```

15. Program P31315; {Tim day cuc dai thoa man tinh chat cho truoc}

```
Uses Crt;
Const   Inp = 'P31314.inp';
        Out = 'P31314.out';
        Nmax = 1000;
Var   f: text;
      n,vt1,vt2,k: integer;
      a,b,c: array [1..Nmax] of real;

Procedure Nhap;
Var i: integer;
Begin
  Assign(f,Inp); Reset(f);
  Readln(f,n);
  For i:=1 to n do Readln(f,a[i],b[i],c[i]);
  Close(f);
End;
```

```
Function Ok(i,j: integer): boolean;
Var t: integer;
Begin
 Ok:=false;
 For t:=i+1 to j do If (a[t]<a[t-1]) or (b[t]<b[t-1]) or (c[t]<c[t-1]) then
exit;
 Ok:=true;
End;

Procedure TimDay;
Var i,j: integer;
Begin
 k:=0;
 For i:=1 to n-1 do
  For j:=n downto i+1 do
    If (Ok(i,j))and(k<j-i+1) then
      begin
          k:=j-i+1;
          vt1:=i;
          vt2:=j;
      end;
End;

Procedure InKq;
Var i: integer;
Begin
 Assign(f,Out); Rewrite(f);
 If k=0 then Writeln(f,'No solution')
   Else
      begin
          Writeln(f,k);
          For i:=vt1 to vt2 do Writeln(f,a[i]:0:2,' ',b[i]:0:2,' ',c[i]:0:2);
      end;
 Close(f);
End;

BEGIN
 Nhap;
 TimDay;
 InKq;
END.
```

Bài 14. Các thuật toán tối ưu

1. Program P31401; {Tim day don dieu tang cuc dai}

```
Uses Crt;
Const- Nmax=5000;
        Inp='P31401.inp';
        Out='P31401.out';
Var F: text;
     N,Max.DanhDau: integer;
     A,Pre,L: array [1..NMax] of integer;

Procedure Nhap;
Var i: integer;
Begin
  Assign(F,inp); Reset(F);
  Readln(F.N);
  For i:=1 to N do Read(F,A[i]);
  Close(f);
End;
Procedure XuLy;
Var i,j: integer;
Begin
{ Xay dung do dai cac day don dieu tang }
  Fillchar(Pre,sizeof(Pre),0);
  For i:=1 to N do L[i]:=1;
  For i:=2 to N do
  For j:=1 to i-1 do
    If (A[j]<=A[i])and(L[i]<L[j]+1) then
       begin
            Pre[i]:=j;
            L[i]:=L[j]+1;
       end;
  { Xac dinh day don dieu tang nhieu phan tu nhat }
  Max:=0;
  For i:=1 to N do
    If L[i]>Max then
       begin
            Max:=L[i];
            DanhDau:=i;
       end;
  { Danh dau cac phan tu thuoc day da xac dinh }
```

```
 Fillchar(L,sizeof(L),0);
 While DanhDau<>0 do
   begin
        L[DanhDau]:=1;
        DanhDau:=Pre[DanhDau];
   end;
End;
Procedure InKq;
Var i,k: integer;
Begin
 Assign(F,Out); Rewrite(F);
 Writeln(F,Max);
 For i:=1 to N do If L[i]=1 then Write(F,A[i],' ');
 Close(F);
End;

BEGIN
 Clrscr;
 Nhap;
 XuLy;
 InKq;
END.
```

3. Program P31403; {Tim tong con co gia tri nho nhat}

```
Uses Crt;
Const  Inp ='P31403.inp';
       Out ='P31403.out';
       Nmax=1000;
Var  F: text;
     N: integer;
     A,Tong: array [1..Nmax] of real;
     Dau,Cuoi: integer;
     Min: real;

Procedure Nhap;
Var i: integer;
Begin
 Assign(F,Inp); Reset(F);
 Readln(F,N);
 For i:=1 to N do Read(f,A[i]);
 Close(F);
End;
```

```
Procedure TimDay;
Var i,j: integer;
Begin
 Tong[1]:=A[1];
 For i:=2 to N do Tong[i]:=Tong[i-1]+A[i];
 Min:=Maxint;
 For i:=1 to N-1 do
  For j:=i+1 to N do
    If Tong[j]-Tong[i]+A[i]<Min then
      begin
          Min:=Tong[j]-Tong[i]+A[i]:
          Dau:=i; Cuoi:=j;
      end;
End;
Procedure InKq;
Var i: integer;
Begin
 Assign(f,Out); Rewrite(f);
 Writeln(f,Min:0:2);
 For i:=Dau to Cuoi do Write(f,A[i]:0:2,' ');
 Close(f);
End;

BEGIN
 Clrscr;
 Nhap;
 TimDay;
 InKq;
END.
```

5. Program P31405; {Lap cach goi dien thoai cua N nguoi}

```
Uses Crt;
Const   fg ='P31405.out';
        max=1000;
Var  f: text;
     n:integer;
     tt,tt1: array [0..max] of byte;
     sp,sl : integer;
     ketthuc: boolean;
```

```
Procedure Nhap;
Begin
  Write('N = '); Readln(n);
End;
Procedure ChuanBi;
Var x: longint;
Begin
  Fillchar(tt,sizeof(tt),1);
  x:=1;
  sp:=0;
  While x<n do
    begin
        inc(sp);
        x:=x shl 1;
    end;
End;
Procedure Tim1(Var i: integer);
Var ii: integer;
Begin
  i:=0;
  For ii:=1 to n do
    If tt1[ii]=1 then
      begin
            i:=ii;
            exit;
      end;
End;
Procedure Tim0(Var i: integer);
Var ii:   integer;
Begin
  i:=0;
  For ii:=1 to n do
    If tt[ii]=0 then
      begin
            i:=ii;
            exit;
      end
End;

Function ThuGhepTiepDuoc: boolean;
Var i,j: integer;
Begin
```

```
  ThuGhepTiepDuoc:=false;
  Tim1(i);
  tt1[i]:=0;
  Tim1(j);
  tt1[j]:=0;
  If (i=0)or(j=0) then exit;
  Write(f,'(',i,',',j,') ');
  tt[j]:=0;
  dec(sl);
  ThuGhepTiepDuoc:=true;
End;

Procedure ThongTinDau;
Begin
  tt1:=tt;
  Repeat
     ketthuc:=not ThuGhepTiepDuoc;
  Until ketthuc;
End;
Function TraoTiepDuoc: boolean;
Var i,j: integer;
Begin
  TraoTiepDuoc:=false;
  Tim1(i);
  tt1[i]:=0;
  Tim0(j);
  If (i=0)or(j=0) then exit;
  tt[j]:=1;
  inc(sl);
  Write(f,'(',i,',',j,') ');
  TraoTiepDuoc:=true;
End;
Procedure ThongTinCuoi;
Begin
  tt1:=tt;
  Repeat
     ketthuc:=not TraoTiepDuoc;
  Until ketthuc;
End;

Procedure LapKeHoach;
Begin
```

```
 Assign(f,fg); Rewrite(f);
 Writeln(f,sp*2);
 sl:=n;
 While sl<>1 do
    begin
        ThongTinDau;
        Writeln(f);
    end;
 While sl<>n do
    begin
         ThongTinCuoi;
          Writeln(f);
    end;
 Close(f);
End;

BEGIN
 Clrscr;
 Nhap;
 ChuanBi;
 LapKeHoach;
END.
```

```
7. Program P31407; {Doi cho N nguoi quanh ban tron}
Uses Crt;
Const    Out = 'P31407.out';
         Nmax= 1000;
Var f: text;
     n,Tong: integer;
     cs,a : array [1..Nmax] of integer;

Procedure Nhap;
Begin
  Write('N = '); Readln(n);
End;

Procedure ChuanBi;
Var i: integer;
Begin
  For i:=1 to n do cs[i]:=i;
  a:=cs;
```

```
  Tong:=0;
End;

Procedure Doi(Var a,b: integer);
Var t: integer;
Begin
 t:=a; a:=b; b:=t
End;

Procedure TimViTri(i: integer; Var j: integer);
Var ii: integer;
Begin
 For ii:=1 to n do
  If cs[ii]=i then
   begin
       j:=ii;
       exit;
   end;
End;

Procedure DoiTrai(i: integer);
Var j: integer;
Begin
 j:=i-1;
 If j=0 then j:=n;
 Writeln(f,a[j],' ',a[i]);
 Doi(a[i],a[j]);
 Doi(cs[i],cs[j]);
End;

Procedure DoiSangTrai(vt,k: integer);
Var i: integer;
Begin
 For i:=1 to k do
  begin
      DoiTrai(vt);
      dec(vt);
      If vt=0 then vt:=n;
  end;
End;

Procedure DoiPhai(i: integer);
```

```
Var j: integer;
Begin
 j:=i+1;
 If j=n+1 then j:=1;
 Writeln(f,a[i],' ',a[j]);
 Doi(a[i],a[j]);
 Doi(cs[i],cs[j]);
End;

Procedure DoiSangPhai(vt,k: integer);
Var i: integer;
Begin
 For i:=1 to k do
   begin
       DoiPhai(vt);
       Inc(vt);
       If vt=n+1 then vt:=1;
    end;
End;

Procedure TimCachDoi(i: integer);
Var j: integer;
Begin
 TimViTri(i,j);
 If i-1<n-i then DoiSangTrai(j,i-1)
   Else DoiSangPhai(j,n-i);
End;

Function Min(a,b: integer): integer;
Begin
 If a<b then Min:=a Else Min:=b
End;

Procedure SapXep;
Var i: integer;
Begin
  Assign(f,Out); Rewrite(f);
  For i:=2 to n-1 do Inc(Tong,Min(i-1,n-i));
  Writeln(f,Tong);
  For i:=2 to n-1 do TimCachDoi(i);
  Close(f);
 End;
```

```
BEGIN
 Clrscr;
 Nhap;
 ChuanBi;
 SapXep;
END.
```

9. Program P31409; {Danh so vao luoi o vuong}

```
Uses Crt;
Const   Out='KQ.txt';
        MNmax=150;
Var  f: text;
     m,n: integer;
     a : array [0..MNmax,0..MNmax] of integer;

Procedure Nhap;
Begin
 Write('So hang M = '); Readln(m); Dec(m);
 Write('So cot N = '); Readln(n); Dec(n);
End;

Procedure GhiSo;
Var i,j: integer;
Begin
 For i:=0 to m do
   For j:=0 to n do a[i,j]:=i xor j;
End;

Procedure InKq;
Var i,j: integer;
Begin
 Assign(f,Out); Rewrite(f);
 For i:=m downto 0 do
   begin
       For j:=0 to n do Write(f,a[i,j]:4);
       Writeln(f);
   end;
 Close(f);
End;
```

```
BEGIN
 Clrscr;
 Nhap;
 GhiSo;
 InKq;
END.
```

11. Program P31411; {Sinh ngau nhien cac phan hoach}

```
Uses Crt;
Const   fg = 'PHOACH.OUT';
        max = 100;
Var  n: integer;
     a: array [0..max+1,0..max+1] of byte;
    dtcon: longint;
     sh: integer;

Procedure Nhap;
Begin
 Write('Sinh phan hoach cho hinh vuong canh ??? ');
 Readln(n);
End;

Procedure Chuanbi;
Var i,j: integer;
Begin
 FillChar(a,SizeOf(a),1);
 For i:=1 to n do
  For j:=1 to n do a[i,j]:=0;
 dtcon:=n*n;
 sh:=0;
End;

Procedure Sinhngaunhienchunhat(k,i,j: integer);
Var ii,jj: integer; ngang,doc: integer;
Begin
 ii:=1; jj:=1;
 While a[i+ii-1,j]=0 do Inc(ii);
 Dec(ii);
 While a[i,j+jj-1]=0 do Inc(jj);
 Dec(jj);
```

```
 doc:=Random(ii)+1;
 ngang:=Random(jj)+1;
 Dec(dtcon,ngang*doc);
 For ii:=1 to doc do
   For jj:=1 to ngang do a[i+ii-1,j+jj-1]:=k;
End;

Procedure Ghinhan;
Var f: text; i,j: integer;
Begin
 Assign(f,fg); Rewrite(f);
 For i:=1 to n do
   begin
        For j:=1 to n do Write(f,a[i,j]:4);
        Writeln(f);
    end;
 Close(f);
End;

Procedure Sinh;
Var i,j: integer;
Begin
 For i:=1 to n do
   For j:=1 to n do
    If a[i,j]=0 then
      begin
           Inc(sh);
          Sinhngaunhienchunhat(sh,i,j);
      end;
End;

BEGIN
 Randomize;
 Clrscr;
 Nhap;
 Chuanbi;
 Sinh;
 Ghinhan;
END.
```

```
13. {$R+,B-}
Program P31413; {Tro choi boc soi}
Uses Crt;
Var dong: array [1..100] of Integer;
     dap : char;
     n : integer;

Procedure Nhap;
Var i: integer; x,y: integer;
Begin
 n:=3;
 For i:=1 to n do
   begin
         Write('Dong ',i,' co ');
         Readln(dong[i]);
     end;
 Write('Ban thich di truoc khong C/K ');
 x:=wherex; y:=wherey;
 Repeat
     gotoxy(x,y);
     Clreol;
     Readln(dap);
     dap:=upcase(dap);
 Until (dap='C')or(dap='K');
End;

Procedure Nguoidi;
Var i,j: integer; x,y: integer;
Begin
 For i:=1 to n do Write(dong[i],' ');
 Writeln;
 Write('Ban boc o dong ');
 x:=wherex; y:=wherey;
 Repeat
     gotoxy(x,y);
     Clreol;
     Readln(i);
 Until (i in [1..n])and(dong[i]>0);
 Write('Ban boc may vien ');
 x:=wherex; y:=wherey;
 Repeat
```

```
    gotoxy(x,y);
    clreol;
    Readln(j);
  Until j in [1..dong[i]];
  Dec(dong[i],j);
End;

Procedure Boc(p: integer; Var ok: boolean);
Var k,i: integer;
Begin
  k:=0;
  For i:=1 to n do If i<>p then k:=k xor dong[i];
  If k<dong[p] then
    begin
        Ok:=true;
        Write('May boc ',dong[p]-k,' vien o dong ',p);
        dong[p]:=k;
    end;
End;

Procedure Bochuhoa;
Var i: integer;
Begin
  For i:=1 to 3 do
    If dong[i]>0 then
      begin
          Dec(dong[i]);
          exit;
      end;
End;

Procedure Maydi;
Var  Ok: boolean; i: integer;
Begin
  Ok:= false;
  For i:=1 to n do If not Ok then Boc(i,ok);
  If not Ok then Bochuhoa;
End;

Function Het: boolean;
Var k,i: integer;
Begin
```

```
 k:=0;
 For i:=1 to n do k:=k+dong[i];
 het:=k=0;
End;

Procedure Choi;
Begin
 If dap='C' then nguoidi;
 If het then
   begin
         Writeln;
         Writeln('Ban thang roi do');
         exit;
   end;
 While not Het do
    begin
         maydi;
         If het then
           begin
                 Writeln;
                 Writeln('Ban thua roi do');
                 exit;
           end;
         Readln;
         Clrscr;
         Nguoidi;
         If het then
           begin
                Writeln;
                Writeln('Ban thang rui do');
           end;
    end;
End;

BEGIN
 Clrscr;
 Nhap;
 Choi;
 Readln;
END.
```

```
15. Program P31415; {Bai toan hai dong soi tong quat}
{ So soi nhieu nhat : 1000 vien }
Uses Crt;
Const   max = 10000;
        maxc = 100;
Var c: integer;
    s1,s2: integer;
    ddlen,ddcheo: array [-maxc-1..max] of byte;
    luu : array [0..max] of integer;
    socan : integer;
    maxcheo: integer;
    ketthuc: boolean;

Procedure Nhap;
Begin
  Write('So soi o dong thu 1: '); Readln(s1);
  Write('So soi o dong thu 2: '); Readln(s2);
  Write('Hang so gioi han ³k - l³: '); Readln(c);
End;

Procedure Chuanbi;
Var i: integer;
Begin
  FillChar(ddlen,SizeOf(ddlen),0);
  FillChar(ddcheo,SizeOf(ddcheo),0);
  socan:=s1;
  If socan<s2 then socan:=s2;
  For i:=1 to socan do luu[i]:=max+1;
  maxcheo:=0;
End;

Function khongdau: integer;
Var ii: integer;
Begin
  ii:=0;
  While (ddlen[ii]<>0) or (ddcheo[ii]<>0) do Inc(ii);
  khongdau:=ii;
End;

Procedure Tinhgiatriluu(i: integer);
Var ii,jj: integer;
Begin
```

```
 ii:=khongdau;
 luu[i]:=ii;
 ddlen[ii]:=1;
 For jj:= -c to c do ddcheo[ii+jj]:=1;
 If maxcheo<ii+c then maxcheo:=ii+c;
End;

Procedure Tangdanhdau;
Var i: integer;
Begin
 Inc(maxcheo);
 For i:=maxcheo downto -c do ddcheo[i]:=ddcheo[i-1];
End;

Procedure danhdau;
Var i: integer;
Begin
 For i:=0 to socan do
   begin
        Tinhgiatriluu(i);
        Tangdanhdau;
    end;
End;

Procedure boc(dong,giam: integer);
Begin
 Writeln('Boc ',giam,' vien tu dong thu ',dong);
 If dong=1 then Dec(s1,giam) Else Dec(s2,giam);
End;

Procedure didai;
Begin
 If s1<s2 then boc(2,1) Else boc(1,1);
End;

Procedure Ditoiuu;
Var ii: integer;
Begin
 If s1>luu[s2] then
   begin
        boc(1,s1-luu[s2]);
        exit;
    end;
```

```
 If s2>luu[s1] then
    begin
        boc(2,s2-luu[s1]);
        exit;
    end;
 For ii:=s1-1 downto 0 do
   If (abs((s1-ii)-(s2-luu[ii]))<=c)and(s2>=luu[ii]) then
      begin  boc(1,s1-ii); boc(2,s2-luu[ii]);  end;
End;

Procedure Kiemtra;
Begin
 Ketthuc:= (s1=0) and (s2=0);
End;

Procedure Maychoi;
Begin
 If s1=luu[s2] then didai Else ditoiuu;
End;
Procedure Nguoiboc1dong;
Var dong,soi: integer; ok: boolean;
Begin
 Repeat
     Write('Ban boc dong nao ???[1/2] '); Readln(dong);
     Write('Ban boc may vien ??? '); Readln(soi);
      ok:=false;
      If dong=1 then If s1<soi then ok:=true Else Dec(s1,soi)
         Else If s2<soi then ok:=true Else Dec(s2,soi);
      If ok then writeln('Boc lai ... ');
 Until not ok;
End;

Procedure Nguoiboc2dong;
Var soi: integer;
Begin
 Write('Ban boc may vien o dong 1 ???'); Readln(soi);
 Dec(s1,soi);
 Write('Ban boc may vien o dong 2 ???'); Readln(soi);
 Dec(s2,soi);
End;

Procedure Nguoichoi;
```

```
Var sl: integer;
Begin
  Write('Ban boc 1 dong hay 2 ???[1/2] '); Readln(sl);
  If sl=1 then nguoiboc1dong Else nguoiboc2dong;
End;

Procedure hienthi;
Begin
  Writeln('So soi hien tai: (1): ',s1,' (2): ',s2);
End;
Procedure Choi;
Begin
  Repeat
      nguoichoi;
      kiemtra;
      hienthi;
      If ketthuc then exit;
      maychoi;
      kiemtra;
      hienthi;
  Until ketthuc;
End;

BEGIN
  Clrscr;
  Nhap;
  Chuanbi;
  Danhdau;
  Choi;
  Readln;
END.
```

Bài 15. Các bài toán thi học sinh giỏi

```
1. Program P31501; {IOI 94}
Const Test = true;
Var inp,out: text;

Procedure Init;
Begin
 If Test then Writeln('IOI"94 - Day 1 - Problem 1: The Triangle');
 Assign(inp,'input.txt'); Reset(inp);
 Assign(out,'output.txt'); ReWrite(out);
 If Test then Writeln('Initialized')
End;
Procedure Fini;
Begin
 Close(inp); Close(out)
End;

Const MaxN = 100;
Type   index = 1..MaxN;
       number = 0..99;
Var  N: index;
     T: array [index,index] of number;
     m: integer;
Procedure ReadInput;
Var r,p: index;
Begin
 Readln(inp,N);
 If Test then Writeln('Number of rows is ',N:1);
 For r:=1 to N do
   begin
       For p:=1 to r do Read(inp,T[r,p]);
       Readln(inp);
   end;
 If Test then Writeln('All input Read')
End;
Procedure ComputeAnswer;
Function Max(x,y: integer): integer;
Begin
 If x>y then Max:=x Else Max:=y
 End;
```

```
Function MaxRS(r,p: index): integer;
Begin
 If r=N then MaxRS:=T[r,p]
   Else MaxRS:=T[r,p]+Max(MaxRS(r+1,p),MaxRS(r+1,p+1))
End;
Begin
 m:=MaxRS(1,1)
End;
Procedure WriteOutput;
Begin
 Writeln(out,m:1);
 If Test then Writeln('The maximum route sum is ',m:1)
End;

BEGIN
 Init;
 ReadInput;
 ComputeAnswer;
 WriteOutput;
 Fini
END.
```

3. Program P31503; {IOI 94}

```
Const Test = true;
Var inp,out: text;
Procedure Init;
Begin
 If Test then Writeln('IOI"94 - Day 1 - Problem 3: The Primes');
 Assign(inp,'input.txt'); Reset(inp);
 Assign(out,'output.txt'); ReWrite(out);
 If Test then Writeln('Initialized')
End;
Procedure Fini;
Begin
 Close(inp); Close(out);
End;

Type  digit = 0..9;
      index = 1..5;

Var  ds: integer;
     tld: digit;
```

```
      S: array [index, index] of digit;

Procedure ReadInput;
Begin
  Readln(inp,ds);
  If Test then Writeln('Digit sum is ',ds:1);
  Readln(inp,tld);
  If Test then Writeln('Digit in top left-hand corner is ',tld:1);
End;

Type   number = Record
                 d: array [index] of digit;
                 v: integer;
               End;

Var   n: integer;
      prime: array [0..800] of number;
      first,last: array [digit] of integer;

Procedure ReadPrimes;
Var  primes: text;
      p,s: integer;
      i: index;
      f: digit;
Begin
  Assign(primes,'primes-5.dat'); Reset(primes);
  For n:=1 to 9 do begin first[n]:=-1; last[n]:=-2; end;
  n:=0;
  While not eof(primes) do
    begin
        Readln(primes,p,s);
        If s=ds then With prime[n] do
          Begin
              v:=p;
              For i:=5 downto 1 do begin d[i]:=p mod 10; p:=p div 10; end;
              If first[d[1]]=-1 then first[d[1]]:=n;
              last[d[1]]:=n;
              Inc(n);
          End;
    end;
  If Test then
```

```
  begin
      Writeln('Number of 5-digit primes with digit sum ',ds:1,' is ',n:1);
      Writeln('Number of these primes with first digit f:');
      Write(' f = ');
      For f:=1 to 9 do Write(f:4);
      Writeln; Write(' # = ');
      For f:=1 to 9 do Write(last[f]+1-first[f]:4);
      Writeln;
  end;
End;

Function IsPrime(w: integer): boolean;
Var i,j,h: integer;
Begin
 i:=0; j:=n;
 While i<>pred(j) do
  begin
      h:=(i+j) div 2;
      If prime[h].v<=w then i:=h Else j:=h
  end;
 IsPrime:=(prime[i].v=w);
End;

Procedure WriteSquare(Var ff: text);
Var r,c: index;
Begin
 For r:=1 to 5 do
  begin
      For c:=1 to 5 do Write(ff,S[r,c]:1);
      Writeln(ff)
  end;
 Writeln(ff);
End;

Var solutions: integer;

Procedure WriteSolution;
Begin
 WriteSquare(out);
 Inc(solutions);
 If Test then
   begin
```

```
        Writeln('Solution ',solutions:1);
        WriteSquare(output)
    end;
End;

Procedure H1; Forward;
Procedure H2; Forward;
Procedure H3; Forward;
Procedure H4; Forward;
Procedure H5; Forward;
Procedure V1; Forward;
Procedure V2; Forward;
Procedure V3; Forward;
Procedure V4; Forward;
Procedure V5; Forward;
Procedure D1; Forward;
Procedure D2; Forward;

Procedure ComputeAnswer;
Begin
 S[1,1]:=tld;
 solutions:=0;
 H1;
 If Test then Writeln('Number of solutions = ',solutions:1)
End;

Procedure H1;
Const R=1;
Var i: integer; c: index;
Begin
 For i:=first[tld] to last[tld] do with prime[i] do
 If d[2]<>0 then
  If d[3]<>0 then
   If d[4]<>0 then
    begin
        For c:=2 to 5 do S[R,c]:=d[c];
        V1;
    end;
End;

Procedure V1;
Const C=1;
```

```
Var i: integer; r: index;
Begin
 For i:=first[tld] to last[tld] do With prime[i] do
  If d[2]<>0 then
   If d[3]<>0 then
    If d[4]<>0 then
      begin
            For r:=2 to 5 do S[r,C]:=d[r];
            D2;
        end;
End;
Procedure D2;
Var i: integer;
Begin
 For i:=first[S[5,1]] to last[S[5,1]] do With prime[i] do
  If d[5]=S[1,5] then
   begin
         S[4,2]:=d[2]; S[3,3]:=d[3]; S[2,4]:=d[4];
         H2;
   end;
End;
Procedure H2;
Const R = 2;
Var i: integer;
Begin
 For i:=first[S[R,1]] to last[S[R,1]] do With prime[i] do
  If d[4]=S[R,4] then
    Begin
         S[R,2]:=d[2]; S[R,3]:=d[3]; S[R,5]:=d[5];
         V2;
     end;
End;

Procedure V2;
Const C = 2;
Var i: integer;
Begin
 For i:=first[S[1,C]] to last[S[1,C]] do With prime[i] do
  If d[2]=S[2,C] then
   If d[4]=S[4,C] then
     begin
            S[3,C]:=d[3]; S[5,C]:=d[5];
```

```
            H3;
        end;
End;

Procedure H3;
Const R = 3;
Var i: integer;
Begin
 For i:=first[S[R,1]] to last[S[R,1]] do With prime[i] do
 If d[2]=S[R,2] then
  If d[3]=S[R,3] then
    begin
          S[R,4]:=d[4]; S[R,5]:=d[5];
          V3;
      end;
End;

Procedure V3;
Const C = 3;
Var i: integer;
Begin
 For i:=first[S[1,C]] to last[S[1,C]] do With prime[i] do
  If d[2]=S[2,C] then
    If d[3]=S[3,C] then
     begin
          S[4,C]:=d[4]; S[5,C]:=d[5];
          H4;
      end;
End;

Procedure H4;
Const R = 4;
Var i: integer;
Begin
 For i:=first[S[R,1]] to last[S[R,1]] do With prime[i] do
  If d[2]=S[R,2] then
   If d[3]=S[R,3] then
    begin
          S[R,4]:=d[4]; S[R,5]:=d[5];
          V4;
      end;
End;
```

```
Procedure V4;
Const C = 4;
Var d,w: integer; r: index;
Begin
 d:=ds; w:=0;
 For r:=1 to 4 do
   begin
        d:=d-S[r,C];
         w:=10*w+S[r,C]
    end;
 If odd(d)and(0<=d)and(d<=9) then
    If IsPrime(10*w+d) then
      begin
            S[5,C]:=d;
            H5;
       end;
End;
Procedure H5;
Const R = 5;
Var d,w: integer; c: index;
Begin
 d:=ds; w:=0;
 For c:=1 to 4 do
   begin
        d:=d-S[R,c];
        w:=10*w+S[R,c]
   end;
 If odd(d)and(0<=d)and(d<=9) then
   If IsPrime(10*w+d) then
     begin
            S[R,5]:=d;
            V5;
       end;
End;

Procedure V5;
Const C = 5;
Var w: integer; r: index;
Begin
 w:=0;
 For r:=1 to 5 do w:=10*w+S[r,C];
 If IsPrime(w) then D1;
```

```
End;
Procedure D1;
Var w: integer; rc: index;
Begin
  w:=0;
  For rc:=1 to 5 do w:=10*w+S[rc,rc];
  If IsPrime(w) then WriteSolution;
End;

BEGIN
  Init;
  ReadInput;
  ReadPrimes;
  ComputeAnswer;
  Fini
END.
```

5. Program P31505; {Da bong thieu tinh than thi dau}

```
Uses Crt;
Const  Inp = 'Inp.txt';
       Out = 'Out.txt';
       Nmax= 21;
       Diem: array [1..3] of shortint = (0,1,3);
       DiemNguoc: array [1..3] of shortint = (0,3,1);
Var f: text;
    n: byte;
    ThuTu: longint;
    st: string;
    a,Luu: array [1..Nmax,1..Nmax] of shortint;

Procedure Chuyen(i: byte);
Var j,k: byte; so: string; c: integer;
Begin
  st:=st+' ';
  k:=1;
  For j:=1 to n+1 do
    begin
      While st[k]=' ' do inc(k);
      so:='';
      While st[k]<>' ' do
        begin
          so:=so+st[k];
```

```
            inc(k);
          end;
      If so='?' then a[i,j]:=-1 Else val(so,a[i,j],c);
    end;
End;
Procedure Nhap;
Var i: byte;
Begin
  Assign(f,inp); Reset(f);
  Readln(f,n);
  For i:=1 to n do
    begin
        Readln(f,st):
        Chuyen(i);
    end;
  Close(f);
End;
Procedure LoaiBot1;
Var i,j: byte;
Begin
  For i:=1 to n do
    For j:=1 to n do
      If a[i,j]=-1 then
        Case a[j,i] of
              0: a[i,j]:=0;
              1: a[i,j]:=3;
              3: a[i,j]:=1;
            End;
End;

Procedure LoaiBot2;
Var i,j: byte;
Begin
  Luu:=a;
  For i:=1 to n do
   If a[i,n+1]<>-1 then
      For j:=1 to n do If a[i,j]<>-1 then a[i,n+1]:=a[i,n+1]-a[i,j];
  ThuTu:=0;
End;
Procedure GhiNhan;
Var i,j,s: integer;
Begin
```

```
  Inc(ThuTu);
  Writeln(f,ThuTu);
  For i:=1 to n do
   begin
       s:=0;
       For j:=1 to n do
        begin
             Write(f,a[i,j]:3);
             s:=s+a[i,j];
         end;
       Writeln(f,s:3);
    end;
End;

Procedure Try(i,j: byte);
Var k: byte;
Begin
 If i=n then
  begin
       GhiNhan;
       exit;
  end;
 If j=n+1 then
  begin
       If a[i,j]>0 then exit;
       try(i+1,i+2);
       exit;
  end;
 If Luu[i,j]<>-1 then
   begin
       try(i,j+1);
       exit;
   end;
 For k:=1 to 3 do
  If a[i,n+1]>=0 then
    begin
       If Diem[k]<=a[i,n+1] then
        begin
             a[i,j]:=Diem[k];
             a[j,i]:=DiemNguoc[k];
             a[i,n+1]:=a[i,n+1]-Diem[k];
             a[j,n+1]:=a[j,n+1]-DiemNguoc[k];
```

```
            try(i,j+1);
            a[i,n+1]:=a[i,n+1]+Diem[k];
            a[j,n+1]:=a[j,n+1]+DiemNguoc[k];
          end;
    end
  Else
    begin
        a[i,j]:=Diem[k];
        a[j,i]:=DiemNguoc[k];
        a[i,n+1]:=a[i,n+1]-Diem[k];
        a[j,n+1]:=a[j,n+1]-DiemNguoc[k];
        try(i,j+1);
        a[i,n+1]:=a[i,n+1]+Diem[k];
        a[j,n+1]:=a[j,n+1]+DiemNguoc[k];
    end;
End;

Procedure XuLy;
Begin
  LoaiBot1;
  LoaiBot2;
  Assign(f,out); ReWrite(f);
  Try(1,1);
  If ThuTu=0 then GhiNhan;
  Close(f);
End;

BEGIN
  Clrscr;
  Nhap;
  XuLy;
END.
```

7. Program P31507;

```
Uses Crt;
Const  Fi='Input7.txt';
       Fo='Output7.txt';
       M : array [0..7] of integer = (1,1,2,6,24,120,720,5040);
       Mx: array [1..5] of integer = (0,-1,0,1,0);
       My: array [1..5] of integer = (0,0,1,0,-1);
       Ex: array [1..5] of integer = (0,1,0,-1,0);
       Ey: array [1..5] of integer = (0,0,-1,0,1);
```

```
Type  mang = array [1..4,1..4] of integer;
        mang2 = array [0..8] of byte;
        mang4 = array [1..40321] of shortint;
Var  f: text;
      MA,Md,Temp: mang;
      B : mang2;
      Q : mang4;
      Q1 : ^mang4;
      DD : array [1..16] of integer;
     dau,cuoi,di,sl,Time,ct: longint;

Procedure Nhap;
Var i,j: integer;
Begin
  Assign(f,fi); Reset(f);
  For i:=1 to 4 do
    begin
        For j:=1 to 4 do Read(f,md[i,j]);
        Readln(f);
    end;
  Close(f);
End;

Procedure Duoi(a: mang; Var B: mang2);
Var i,j: integer;
Begin
  Fillchar(b,sizeof(b),0);
  For i:=1 to 2 do
    For j:=1 to 4 do B[(i-1)*4+j]:=A[i,j];
End;

Procedure co(Var a: mang; B: mang2);
Var i,j: integer;
Begin
  Fillchar(a,sizeof(a),0);
  For i:=1 to 2 do
    For j:=1 to 4 do A[i,j]:=B[(i-1)*4+j];
End;

Function Vitri(A: mang2): Longint;
```

```
Var i,j: integer; Dx: mang2; T: Longint;
Begin
 Fillchar(dx,sizeof(Dx),0);
 t:=1;
 For i:=1 to 8 do
  begin
      For j:=A[i]-1 downto 1 do
      If Dx[j]=0 then T:=T+M[8-i];
      dx[A[i]]:=1;
  end;
 Vitri:=t;
End;

Procedure thanhbang(p: longint; Var A: mang2);
Var Dx: mang2; i,j: integer; K: integer;
Begin
 Fillchar(dx,Sizeof(dx),0);
 p:=p-1;
 For i:=1 to 8 do
  begin
     k:=p div M[8-i]+1;
     p:=p mod M[8-i];
     j:=1;
     While k>0 do
       begin
            While Dx[j]=1 do Inc(j);
            Inc(j);
            Dec(k);
        end;
     A[i]:=j-1;
     Dx[j-1]:=1;
  end;
End;

Procedure Find(p: integer; Var x,y: integer; a: mang);
Var i,j: integer;
Begin
 For i:=1 to 4 do
  For j:=1 to 4 do
   If A[i,j]=p then
    begin
        x:=i; y:=j;
```

```
            exit;
        end;
End;
Function trong(x,y: integer): boolean;
Begin
  trong:=(x>0)and(x<=2)and(y>0)and(y<=4);
End;

Procedure coc(Var a,b: integer) ;
Var c: integer;
Begin
  c:=a; a:=b; b:=c;
End;

Procedure Write_arr(A: mang);
Var i,j: integer;
Begin
  Inc(sl);
  Writeln(f,sl);
  If Ct=0 then
      begin
          Find(7,i,j,a); A[i,j]:=0;
          Find(8,i,j,a); A[i,j]:=0;
          For i:=3 to 4 do
            For j:=1 to 4 do A[i,j]:=Md[i,j];
      end
     Else
      begin
          For i:=1 to 2 do
            For j:=1 to 4 do A[i+2,j]:=A[i,j]+8;
          For i:=1 to 2 do
            For j:=1 to 4 do A[i,j]:=md[i,j];
          Find(15,i,j,a); A[i,j]:=0;
          Find(16,i,j,a); A[i,j]:=0;
      end;
  For i:=1 to 4 do
    begin
        For j:=1 to 4 do Write(f,A[i,j]: 3);
        Writeln(f);
    end;
End;
```

```
Function Find_Fist: word;
Var i: word;
Begin
 I:=1;
 While (Q[i]<=0)and(i<=40320) do Inc(i);
 Find_fist:=i;
End;

Procedure Write_out1(k: word);
Var  i: word;
     x,y,px,py,slkq: integer;
     Kq: array [1..300] of mang;
Begin
 Fillchar(kq,sizeof(kq),0);
 slkq:=1;
 i:=k;
 thanhbang(k,b);
 co(Ma,b); Kq[slkq]:=Ma;
 Repeat
     thanhbang(k,b);
     co(Ma,b);
     Find(7,x,y,Ma);
     px:=x+ex[abs(Q[k])]; py:=y+ey[abs(Q[k])];
     Coc(MA[x,y],MA[px,py]);
     Find(8,x,y,Ma);
     px:=x+ex[abs(Q1^[k])]; py:=y+ey[abs(Q1^[k])];
     Coc(MA[x,y],MA[px,py]);
     Inc(slkq);
     Kq[slkq]:=Ma;
     duoi(Ma,b);
     k:=vitri(b);
 Until k=di;
 For i:=slkq downto 1 do Write_arr(Kq[i]);
End;

Procedure loang;
Var  k: word;
     i,j,x1,y1,px1,py1,x2,y2,px2,py2,p: integer;
     La: mang;
```

```
Begin
 Fillchar(q,sizeof(q),0);
 Fillchar(q1^,sizeof(q1^),0);
 Duoi(Ma,b);
 Di:=Vitri(b);
 Q1^[di]:=125; Q[di]:=125;
 Repeat
    k:=find_fist;
    If k=40321 then
     begin
         Write('Khong the bien doi duoc ');
         Exit;
     end;
    Q[k]:=-Q[k]; Q1^[k]:=-Q1^[k];
    Thanhbang(k,b);
    Co(Ma,b); la:=Ma;
    Find(7,x1,y1,Ma);
    Find(8,x2,y2,Ma);
    For i:=1 to 5 do
     For j:=1 to 5 do
      If (i=1) xor (j=1) then
       begin
           px1:=x1+Mx[i]; py1:=y1+My[i];
           px2:=x2+Mx[j]; py2:=y2+My[j];
           If trong(px1,py1)and trong(px2,py2) then
            If (MA[px1,py1]<>8)and(MA[px2,py2]<>7) then
              begin
                 coc(MA[x1,y1],MA[px1,py1]);
                 coc(MA[x2,y2],MA[px2,py2]);
                 duoi(Ma,b); k:=vitri(b);
                 If Q[k]=0 then
                   begin
                      Q[k]:=i; Q1^[k]:=j;
                      If (k=1)or(k=2) then
                        begin
                           Write_out1(k);
                           Exit;
                        end;
                   end;
              end;
```

```
            Ma:=la;
        end;
 Until false;
End;

Procedure Init;
Var i,j: integer;
Begin
 Find(0,i,j,md);
 md[i,j]:=15;
 Find(0,i,j,md);
 md[i,j]:=16;
End;

Procedure Find_bigger(k: integer; Var x,y: integer; Var A: mang);
Var i,j: integer;
Begin
 For i:=1 to 4 do
  For j:=1 to 4 do
   If (Dd[A[i,j]]=0) and (a[i,j]<k) then
    begin
        x:=i; y:=j;
        exit;
    end;
 x:=0;
End;

Procedure Write_arr1(A: mang);
Var i,j: integer;
Begin
 Inc(sl);
 Find(15,i,j,a); A[i,j]:=0;
 Find(16,i,j,a); A[i,j]:=0;
 For i:=1 to 4 do
    begin
        For j:=1 to 4 do Write(f,a[i,j]:3);
        Writeln(f);
    end;
 Writeln(f);
End;
```

```
Procedure Move_tr_x(k,x,y: integer; Var a: mang);
Var x1,y1: integer;
Begin
 Find(k,x1,y1,md);
 While x<x1 do
  begin
       coc(A[x1,y1],a[x1-1,y1]);
       Write_arr1(a);dec(x1);
  end;
 While x>x1 do
  begin
       coc(A[x1,y1],a[x1+1,y1]);
       Write_arr1(a);inc(x1);
  end;
End;

Procedure Move_tr_y(k,x,y: integer; Var A: mang);
Var x1,y1: integer;
Begin
 Find(k,x1,y1,a);
 While y>y1 do
   begin
       coc(A[x1,y1],a[x1,y1+1]);
       Write_arr1(a); Inc(y1);
  end;
 While y<y1 do
   begin
       coc(A[x1,y1],a[x1,y1-1]);
       Write_arr1(a); Dec(y1);
  end;
End;

Procedure Move_tr(k,x,y: integer; Var A: mang);
Var i,j,x1,y1: integer;
Begin
 Move_tr_x(k,x,y,a);
```

```
 Move_tr_y(k,x,y,a);
End;

Procedure move_x(p: integer; x,y: integer; Var A: mang);
Var x1,y1,xt,yt: integer;
Begin
 Find(p,x1,y1,a);
 If (x1=x) then exit;
 Find(15,xt,yt,a);
 If x1<=x then
  begin
      Move_tr(15,x1+1,y1,a);
      Move_tr(15,x1,y1,a);
      If x1<=x then
        begin
             Find(15,xt,yt,a); Find(p,x1,y1,a);
             Move_tr(15,x1+1,y1,a); Move_tr(15,x1,y1,a);
         end;
   end
  Else
     Repeat
        Find(p,x1,y1,a); If (x1=x) then exit;
        Find(15,xt,yt,a);
        If xt<x1 then Move_tr(15,x1-1,y1,a)
         Else If y1<>yt then Move_tr(15,x1-1,y1,a)
                Else
                   begin
                       If y1=4 then Move_tr_y(15,xt,y1-1,a)
                         Else Move_tr_y(15,xt,y1+1,a);
                        Find(15,xt,yt,a); Move_tr(15,x1-1,y1,a)
                    end;
        Move_tr(15,x1,y1,a);
     Until false;
End;

Procedure Move_y(p: integer; x,y: integer; Var A: mang);
Var x1,y1,xt,yt: integer;
Begin
 Repeat
     Find(p,x1,y1,a); Find(15,xt,yt,a);
     If y1=y then exit;
```

```
      If y1>y then
       begin
          Find(p,x1,y1,a); If y1=y then exit;
          Find(15,xt,yt,a);
          If x1< 4 then
            begin
                Move_tr_x(15,x1+1,y1,a);
                Move_tr_y(15,x1+1,y1,a);
             end;
          Find(15,xt,yt,a);
          Move_tr_y(15,x1,y1-1,a);Move_tr_y(15,x1,y1-1,a);
          move_tr(15,x1,y1,a);
       end
      Else If y1<y then
            begin
                Find(p,x1,y1,a); If y1=y then exit;
                Find(15,xt,yt,a);
                If x1< 4 then
                  begin
                      Move_tr_x(15,x1+1,y1,a);
                      Move_tr_y(15,x1+1,y1,a);
                  end;
                Find(15,xt,yt,a);
                Move_tr_y(15,x1,y1+1,a);
                Move_tr_y(15,x1,y1+1,a);
                Move_tr(15,x1,y1,a);
            end;
  Until false;
End;
Procedure move_pt(p: integer; x,y: integer; Var A: mang);
Var x1,y1: integer;
Begin
 If A[x,y]=p then exit;
 Move_tr_y(16,x,y,a);
 Move_tr_x(16,x,y,a);
 Find(p,x1,y1,a);
 If x1=x then
   begin
        Move_y(p,x,y+1,a);
        Move_tr(16,x,y+1,a);
        exit;
```

```
    end;
  Move_x(p,x+1,y,a);
  Move_y(p,x+1,y,a);
  Move_tr(16,x+1,y,a);
End;

Procedure Chiamang;
Var i,j,x,y: integer;
Begin
  Fillchar(Dd,sizeof(dd),0);
  For i:=1 to 4 do
    begin
        Find_bigger(7,x,y,md);
        Dd[md[x,y]]:=1;
        Move_pt(md[x,y],1,i,md);
    end;
  For i:=1 to 2 do
    begin
         Find_bigger(7,x,y,md);
         Dd[md[x,y]]:=1;
         Move_pt(md[x,y],2,i,md);
    end;
End;
Procedure laymang;
Var i,j: integer;
Begin
  Fillchar(Ma,sizeof(Ma),0);
  If ct=0 then
    begin
        For i:=1 to 2 do
          For j:=1 to 4 do MA[i,j]:=Md[i,j];
        Find(15,i,j,Ma);
        MA[i,j]:=7;
        Find(16,i,j,Ma);
        MA[i,j]:=8
    end
   Else
       For i:=3 to 4 do
         For j:=1 to 4 do Ma[i-2,j]:=Md[i,j]-6;
```

```
End;
Procedure Solution;
Var i: integer;
Begin
  New(q1);
  Fillchar(q1^,sizeof(q1^),0);
  Assign(f,fo); Rewrite(f);
  Init; Chiamang;
  Move_tr_y(15,2,3,md);
  Move_tr_x(15,2,3,md);
  Move_tr_y(16,2,4,md);
  Move_tr_x(16,2,4,md);
  ct:=0; Laymang; Loang;
  For i:=1 to 4 do Md[1,i]:=i;
  md[2,1]:=5; md[2,2]:=6;
  Move_pt(7,2,3,md);
  Move_pt(8,2,4,md);
  ct:=1;
  Laymang; Loang;
  Writeln(f,sl);
  Close(f);
  Dispose(q1);
End;

BEGIN
  Clrscr;
  time:=meml[0:$46C];
  Writeln('Please wait.');
  Nhap;
  Solution;
  Writeln((meml[0:$46C]-time) /18.2:8:2);
  Readln;
END.
```

9. Program P31509; {PolygonGame}

```
Uses crt;
Const  minVert = 3;
       maxVert = 50;
       antMaxVert = maxVert - 1;
       opSum   = 't';
Type
       polygonT = record
```

```
                vert: array[0..antMaxVert] of char;
                side: array[0..antMaxVert] of integer;
                numVert: minVert..maxVert
               end;
        maxminT = record
                maxVal, minVal: integer
               end;
        memoryT = array[1..maxVert, 0..antMaxVert] of
maxminT;
        scoresT = record
                scor: array[1..maxVert] of integer;
                numVert, highScor, firstVertHS: integer;
               end;
Var  fileIn, fileOut: text;
     polygon: polygonT;
     scores: scoresT;
Procedure ReadPolygon (var F: text; var P: polygonT);
Var  i: integer;
     c: char;
Begin
  with P do
   begin
        readln(F, numVert);
        for i := 0 to numVert - 2 do  read(F, vert[i], side[i], c);
        read(F, vert[numVert - 1], side[numVert - 1]);
    end;
End;
Procedure Evaluate (Op: char; E1, E2: maxminT; var MM:
maxminT);
Var  values: array[1..4] of integer;
     i, indMax, indMin: integer;
Begin
  if Op = opSum then
     begin
          MM.maxVal := E1.maxVal + E2.maxVal;
          MM.minVal := E1.minVal + E2.minVal
      end
   else
     begin
          values[1] := E1.maxVal * E2.maxVal;
```

```
            values[2] := E1.maxVal * E2.minVal;
            values[3] := E1.minVal * E2.maxVal;
            values[4] := E1.minVal * E2.minVal;
            indMax := 1;
            indMin := 1;
            for i := 2 to 4 do
              if values[i] > values[indMax] then indMax := i
                  else if values[i] < values[indMin] then indMin := i;
            MM.maxVal := values[indMax];
            MM.minVal := values[indMin]
        end
End;
Procedure CompScores (var P: polygonT; var M: memoryT);
Var lastVert, line, inf, infRight, dimLeft: integer;
      mmTemp, mmAux: maxminT;
Begin
 lastVert := P.numVert - 1;
 for inf := 0 to lastVert do
   begin
         M[1, inf].maxVal := P.side[inf];
         M[1, inf].minVal := P.side[inf]
    end;
 for inf := 0 to lastVert do
   begin
         infRight := (inf + 1) mod P.numVert;
         if P.vert[infRight] = opSum then
            M[2, inf].maxVal := M[1, inf].maxVal + M[1,
infRight].maxVal
            else  M[2, inf].maxVal := M[1, inf].maxVal * M[1,
infRight].maxVal;
         M[2, inf].minVal := M[2, inf].maxVal
    end;
 for line := 3 to P.numVert do
   for inf := 0 to lastVert do
    begin
         infRight := (inf + 1) mod P.numVert;
         Evaluate(P.vert[infRight], M[1, inf], M[line - 1, infRight],
mmTemp);
         for dimLeft := 2 to line - 1 do
          begin
```

```
            infRight := (inf + dimLeft) mod P.numVert;
            Evaluate(P.vert[infRight], M[dimLeft, inf], M[line -
dimLeft, infRight], mmAux);
            if mmAux.maxVal > mmTemp.maxVal then
                    mmTemp.maxVal := mmAux.maxVal;
            if mmAux.minVal < mmTemp.minVal then
                    mmTemp.minVal := mmAux.minVal
        end;
        M[line, inf] := mmTemp;
    end;
end;
Procedure CompHighScores (var P: polygonT; var S: scoresT);
Var  mem: memoryT;
     i, maxScor, fstVert: integer;
Begin
  CompScores(P, mem);
  S.scor[1] := mem[P.numVert, 0].maxVal;
  maxScor := S.scor[1];
  fstVert := 1;
  for i := 2 to P.numVert do
    begin
      S.scor[i] := mem[P.numVert, i - 1].maxVal;
      if S.scor[i] > maxScor then
        begin
            maxScor := S.scor[i];
            fstVert := i
        end
    end;
  S.numVert := P.numVert;
  S.highScor := maxScor;
  S.firstVertHS := fstVert
End;
Procedure WriteResults (var F: text; var S: scoresT);
Var  i: integer;
Begin
  with S do
    begin
      writeln(F, highScor : 1);
      write(F, firstVertHS : 1);
      for i := firstVertHS + 1 to numVert do
```

```
      if scor[i] = highScor then  write(F, ' ', i : 1);
    writeln(F)
   end
End;

BEGIN
  assign(fileIn, 'POLYGON.IN');
  reset(fileIn);
  ReadPolygon(fileIn, polygon);
  close(fileIn);
  CompHighScores(polygon, scores);
  assign(fileOut, 'POLYGON.OUT');
  rewrite(fileOut);
  WriteResults(fileOut, scores);
  close(fileOut);
END.
```

11. Program P31511; {RACE}

```
CONST  Max_points = 50;
       Max_Arrows = 100;
TYPE   Point_array = ARRAY [0..Max_points-1] OF BOOLEAN;
VAR  input_file,output_file : text;

current_arrow,current_point,number_of_points,number_of_arrows,
      number_unavoidable_points,number_splitting_points :
INTEGER;
      Arrows : ARRAY [0..Max_Arrows-1,0..1] OF INTEGER;
      unavoidable,splitting : Point_array;
PROCEDURE initialisation;
Begin
  Assign(input_file,'input.txt');
  Reset(input_file);
  Assign(output_file;'output.txt');
  Rewrite(output_file);
  number_of_points:=1;
  number_of_arrows:=0;
  number_unavoidable_points:=0;
  number_splitting_points:=0;
  For current_point:=0 to Max_points-1 do
    begin
        splitting [current_point]:=FALSE ;
        unavoidable [current_point]:=FALSE
```

```
   end;
End;
PROCEDURE read_input:
VAR num : Integer;
Begin
 read(input_file,num);
 WHILE NOT (num = -1) DO
    begin
        IF num = -2 THEN INC (number_of_points)
          ELSE
             begin
                  Arrows [number_of_arrows][0] := number_of_points-1;
                  Arrows [number_of_arrows][1] := num;
                  Inc (number_of_arrows)
              end;
        read(input_file,num)
    end;
End;
PROCEDURE write_output;
Begin
 Write(output_file,number_unavoidable_points);
 FOR current_point:=1 TO number_of_points-2 DO
   IF unavoidable[current_point] THEN write(output_file,'
',current_point);
 Writeln(output_file);
 Write (output_file,number_splitting_points);
 FOR current_point:=1 TO number_of_points-2 DO
   IF splitting[current_point] THEN write(output_file,' ',current_point);
End;

PROCEDURE finalisation;
Begin
 Close(input_file);
 Close(output_file);
End;

PROCEDURE compute_results;
VAR reachable : Point_array;

PROCEDURE find_reachable (current_point:INTEGER);
VAR  point:Integer;
       ready:Boolean;
```

```
Begin
 FOR point:=1 TO number_of_points - 1 DO reachable[point]:=FALSE;
 reachable[0]:=TRUE;
 ready:=FALSE;
 WHILE NOT ready DO
    begin
        ready:=TRUE;
        FOR current_arrow:=0 TO number_of_arrows-1 DO
         IF reachable [Arrows[current_arrow,0]] AND NOT reachable
[Arrows[current_arrow,1]]
             AND (Arrows[current_arrow,1]<>current_point) THEN
           begin
                reachable[Arrows[current_arrow,1]]:=TRUE;
                ready:=FALSE
            end;
    end
End;
FUNCTION is_splitting:BOOLEAN;
VAR current_arrow:INTEGER;
      OK:BOOLEAN;
Begin
 current_arrow:=0;
 OK:=TRUE;
 WHILE (current_arrow<number_of_arrows) AND OK DO
  begin
      OK:=reachable [Arrows[current_arrow,0]] ORNOT (reachable
[Arrows[current_arrow,1]]);
      Inc(current_arrow)
   end;
 is_splitting:=OK
End;

Begin
 FOR current_point:=1 TO number_of_points-2 DO
   begin
        find_reachable (current_point);
        IF NOT reachable[number_of_points-1] THEN
            begin
                unavoidable [current_point] := TRUE;
```

```
            Inc (number_unavoidable_points);
            IF is_splitting THEN
              begin
                  splitting[current_point]:=TRUE;
                  Inc (number_splitting_points);
              end;
        end;
  end;
End;

BEGIN
 initialisation;
 read_input;
 compute_results;
 write_output;
 finalisation;
END.
```

13. Program P31513; {StarryNight}

```
Const   maxMap = 100;
        maxMap1 = maxMap - 1;
        maxShapes = 26;
        maxShapes1 = maxShapes - 1;
Type    Pt = record
             x, y: integer
           end;
        StartPts = array[0..7] of Pt;
        Shape = record
                  name: char;
                  start: StartPts;
                  nStars: integer;
                  num: integer;
                end;
Var     map: array[0..maxMap1, 0..maxMap1] of char; {the Map}
        size: Pt;
        minP, maxP: Pt;
        nStars: integer;
        shapes: array[0..maxShapes1] of Shape;
        arrows: array[0..7] of Pt;
Procedure InitGlobals;
Begin
```

```
        arrows[0].x := 1;
        arrows[0].y := 0;
        arrows[1].x := 1;
        arrows[1].y := 1;
        arrows[2].x := 0;
        arrows[2].y := 1;
        arrows[3].x := -1;
        arrows[3].y := 1;
        arrows[4].x := -1;
        arrows[4].y := 0;
        arrows[5].x := -1;
        arrows[5].y := -1;
        arrows[6].x := 0;
        arrows[6].y := -1;
        arrows[7].x := 1;
        arrows[7].y := -1
End;
Procedure st (p: Pt; n: char);
Begin
        map[p.y, p.x] := n
End;

Function gt (p: Pt): char;
Begin
 if p.x < 0 then gt := '1'
   else if p.x >= size.x then gt := '0'
          else if p.y < 0 then gt := '0'
                 else if p.y >= size.y then gt := '0'
                        else gt := map[p.y][p.x]
End;
Procedure Walk (p: Pt; arrow:integer;rotSym:integer;var res: Pt);
Begin
 if arrow = -1 then res := p
   else
      begin
          arrow := (arrow + 2 * (rotSym mod 4)) mod 8;
          if rotSym div 4 = 0 then
             begin
                 res.x := p.x + arrows[arrow].x;
                 res.y := p.y + arrows[arrow].y;
```

```
                end
              else
                begin
                      res.x := p.x + arrows[arrow].y;
                      res.y := p.y + arrows[arrow].x;
                end;
        end
End;

Procedure LoadMap (fname: string);
Var     f: text;
        w, h, i, j: integer;
        p: Pt;
        c: char;
Begin
  assign(f, fname);
  reset(f);
  readln(f, size.x);
  readln(f, size.y);
  for i := 0 to size.y - 1 do
     begin
        for j := 0 to size.x - 1 do
           begin
                p.x := j;
                p.y := i;
                read(f, c);
                st(p, c);
             end;
        readln(f)
     end;
  close(f)
end;
Procedure WriteMap (fname: string);
Var     f: text;
        i, j: integer;
        p: Pt;
Begin
  assign(f, fname);
  rewrite(f);
  for i := 0 to size.y - 1 do
```

```
    begin
        for j := 0 to size.x - 1 do
          begin
                p.x := j;
                p.y := i;
                write(f, gt(p))
          end;
        writeln(f)
    end;
  close(f)
End;
Procedure UpdateMinMax (p: Pt);
Begin
  nStars := succ(nStars);
  if p.x < minP.x then  minP.x := p.x;
  if p.x > maxP.x then maxP.x := p.x;
  if p.y < minP.y then  minP.y := p.y;
  if p.y > maxP.y then maxP.y := p.y
End;
Procedure FirstFill (p: Pt; arrow: integer; name: char);
Var     a: integer;
        np: Pt;
Begin
  Walk(p, arrow, 0, np);
  if (gt(np) <> '0') and (gt(np) <> name) then
    begin
        st(np, name);
        UpdateMinMax(np);
        for a := 0 to 7 do FirstFill(np, a, name);
    end;
End;

Procedure Fill (p: Pt; arrow: integer; name: char);
Var     a: integer;
        np: Pt;
Begin
  Walk(p, arrow, 0, np);
  if (gt(np) <> '0') and (gt(np) <> name) then
    begin
        st(np, name);
```

```
        for a := 0 to 7 do Fill(np, a, name);
    end;
End;

Procedure StartPt (rotSym: integer; name: char; var res: Pt);
Var     p: Pt;
Begin
  Walk(minP, 1, rotSym, p);
  res := minP;
  if p.x < minP.x then  res.x := maxP.x;
  if p.y < minP.y then  res.y := maxP.y;
  while gt(res) <> name do Walk(res, 0, rotSym, res);
End;

Function CompareX (p, q: Pt; arrow: integer; rotSym: integer):
boolean;
Label   99;
Var     a: integer;
        np, nq: Pt;
Begin
  Walk(p, arrow, 0, np);
  Walk(q, arrow, rotSym, nq);
  if gt(np) = '0' then
    begin
        CompareX := gt(nq) = '0';
        goto 99
    end;
  if gt(np) = '1' then
    begin
        CompareX := true;
        goto 99
    end;
  st(np, '1');
  if gt(nq) = '0' then
    begin
        CompareX := false;
        goto 99
    end;
  for a := 0 to 7 do
    if not CompareX(np, nq, a, rotSym) then
```

```
        begin
            CompareX := false;
            goto 99
        end;
  CompareX := true;
  99: End;
Function Compare (p: Pt; var start: StartPts; old: char): boolean;
Label   99;
Var     rotSym: integer;
Begin
  for rotSym := 0 to 7 do
     begin
        if CompareX(p, start[rotSym], -1, rotSym) then
           begin
               Compare := true;
               goto 99
           end
         else   Fill(p, -1, old)
     end;
  Compare := false;
  99: End;

Procedure ProcessMap;
Label   77, 99;
Var     p: Pt;
        k, rotSym, nShapes: integer;
        i, j: integer;
        name: char;
Begin
  nShapes := 0;
  name := 'a';
  for j := 0 to size.y - 1 do
    for i := 0 to size.x - 1 do
       begin
           p.x := i;
           p.y := j;
           if gt(p) = '1' then
             begin
                 minP := p;
                 maxP := p;
```

```
                nStars := 0;
                FirstFill(p, -1, name);
                for k := 0 to nShapes - 1 do
                  if shapes[k].nStars = nStars then
                     if Compare(p, shapes[k].start, name) then
                        begin
                        shapes[k].num := succ(shapes[k].num);
                              Fill(p, -1, shapes[k].name);
                              goto 77;
                        end;
                if nShapes = maxShapes then
                  begin
                        write('Too many shapes');
                         goto 99
                    end;
                shapes[nShapes].name := name;
                shapes[nShapes].nStars := nStars;
                shapes[nShapes].num := 1;
                for rotSym := 0 to 7 do  StartPt(rotSym, name,
shapes[nShapes].start[rotSym]);
                nShapes := succ(nShapes);
                name := succ(name);
            end;
        77:  end;
 99: End;

BEGIN
 InitGlobals;
 LoadMap('STARRY.IN');
 ProcessMap;
 WriteMap('STARRY.OUT');
END.
```

15. Program P31515;{Sort3}

```
Const   MaxN =1000;
Type    ElemType = 1..3;
        ArrayType = Array[1..MaxN] Of ElemType;
        Matrix = Array[ElemType, ElemType] of Integer;
Var  N : Word;
     S : ArrayType;
     Na: Array[ElemType] Of Word;
```

```
      NEP : Matrix;
      First: Matrix;
      NoCh: Word;
      OutFile: Text;
Procedure ReadInput;
Var  InFile: Text;
      i,j: Word;
Begin
  Assign(InFile, 'input.txt'):
  Reset(Infile);
  ReadLn(InFile, N);
  For i:=1 To 3 Do Na[i]:=0;
  For i:=1 To N Do
     Begin
        ReadLn(InFile,S[i]);
        Inc(Na[S[i]]);
      End;
  Close(InFile);
End;

Function Min(X,Y: Word):Word;
Begin
  If X<Y Then Min:=X
   ELse Min:=Y
End;
Procedure Preprocess;
Var i,j,M:Word;
Begin
 For i:=1 To 3 Do
   Begin
      For j:=1 To 3 Do
         Begin
              NEP[i,j]:=0;
              First[i,j]:=0;
         End;
   End;
  For i:=1 To N Do
```

```
 Begin
   If i<=Na[1] Then
     Begin
       If NEP[S[i],1]=0 Then First[S[i],1]:=i;
       Inc(NEP[S[i],1]);
     End
    Else If i<=Na[1]+Na[2] Then
          Begin
            If NEP[S[i],2]=0 Then First[S[i],2]:=i;
            Inc(NEP[S[i],2]);
          End
         Else Begin
                If NEP[S[i],3]=0 Then First[S[i],3]:=i;
                Inc(NEP[S[i],3])
              End;
  End;
 NoCh:= Min(NEP[1,2], NEP[2,1])+ Min(NEP[1,3], NEP[3,1])
         + Min(NEP[2,3], NEP[3,2])+2*Abs(NEP[1,2]-NEP[2,1]);
End;
Procedure Next(i1,i2:Byte);
Begin
 Dec(NEP[i1,i2]);
 If NEP[i1,i2]>0 Then
   Begin
     Repeat
       Inc(First[i1,i2]);
     Until S[First[i1,i2]]=i1;
   End;
End;
Procedure Pairs;
Var M,i,x,y :Word;
Begin
For x:=1 To 3 Do
 For y:=x+1 To 3 Do
   Begin
     M:=Min(NEP[x,y], NEP[y,x]);
     For i:=1 To M Do
```

```
        Begin
            WriteLn(OutFile, First[x,y],' ',First[y,x]);
            Next(x,y); Next(y,x);
        End;
    End;
End;
Procedure Triples;
Var M,i: Word;
Begin
 If NEP[1,2] > 0 Then
   Begin
     M:=NEP[1,2];
     For i:=1 To M Do
       Begin
          WriteLn(OutFile, First[3,1],' ',First[1,2]);
          WriteLn(OutFile, First[1,2],' ',First[2,3]);
          Next(3,1); Next(1,2); Next(2,3);
       End;
   End
  Else Begin
         M:=NEP[2,1];
         For i:=1 To M Do
          Begin
             WriteLn(OutFile, First[2,1],' ',First[1,3]);
             WriteLn(OutFile, First[1,3],' ',First[3,2]);
             Next(2,1); Next(3,2); Next(1,3);
          End;
       End;
End;

BEGIN
 ReadInput;
 Preprocess;
 Assign(OutFile, 'output.txt');
 Rewrite(OutFile);
 WriteLn(OutFile,NoCh);
 Pairs;
 Triples;
 Close(OutFile);
END.
```

MỤC LỤC

LẬP TRÌNH PASCAL – TẬP III

Mã số: 8H786 - DAI

In 1.000 cuốn, khổ 14,5 x 20,5 cm, tại Xí nghiệp in Hà Tây.

Số in: 1001/DAI; Số xuất bản: 04-2006/CXB/194-1860/GD.

In xong và nộp lưu chiểu tháng 10 năm 2006.

TÌM ĐỌC
TỦ SÁCH "TIN HỌC TRONG NHÀ TRƯỜNG"
CỦA NHÀ XUẤT BẢN GIÁO DỤC

1. Một số vấn đề chọn lọc trong môn tin học (tập một, tập hai)
Nguyễn Xuân My (Chủ biên)

2.Tin học dùng cho học sinh tiểu học - Quyển I
Nguyễn Xuân Huy (Chủ biên)

3. Nghề tin học ứng dụng *Bùi Văn Thanh, Ngô Ánh Tuyết*

4. Lập trình Pascal (tập một, tập hai, tập ba)
Bùi Việt Hà

5. Tìm đường trong mê cung *Nguyễn Xuân Huy*

Bạn đọc có thể tìm mua tại các Công ti Sách – Thiết bị trường học ở các địa phương hoặc các Cửa hàng sách của Nhà xuất bản Giáo dục:

*Tại Hà Nội : **25 Hàn Thuyên, 81 Trần Hưng Đạo, 187B Giảng Võ, 23 Tràng Tiền.***

*Tại Đà Nẵng : **15 Nguyễn Chí Thanh.***

*Tại Thành phố Hồ Chí Minh : **104 Mai Thị Lựu, Quận 1.***

8 934980 686300

Giá: 25.500đ