ĐẬU QUANG TUẤN
Cử nhân Toán - Kỹ sư Tin Học

# BẠN SẼ THÀNH THẠO C++

## Chỉ trong 21 ngày

DÙNG CHO HỌC SINH - KỸ THUẬT VIÊN - SINH VIÊN

- NGHIÊN CỨU PHẦN LÝ THUYẾT TRONG 14 NGÀY
- ĐỌC VÀ HIỂU CÁC BÀI TẬP MẪU TRONG 7 NGÀY

**CUNG CÁCH TRÌNH BÀY :**

Ngắn gọn và dễ hiểu
Hướng dẫn chi tiết và rõ ràng
Các chương trình mẫu thực tế & đa dạng

NHÀ XUẤT BẢN TỔNG HỢP TP. HỒ CHÍ MINH 2005

**Đậu Quang Tuấn**
Cử nhân toán – Kỹ sư tin học

# Bạn sẽ thành thạo

# C++

# *Chỉ trong* 21 *ngày*

*(Dùng cho học sinh, kỹ thuật viên và sinh viên)*

- Nghiên cứu phần lý thuyết trong 14 ngày
- Đọc và hiểu các bài tập mẫu trong 7 ngày

*Cung cách trình bày:*

- *Ngắn gọn và dễ hiểu.*
- *Hướng dẫn chi tiết và rõ ràng.*
- *Các chương trình mẫu thực tế và đa dạng.*

**NHÀ XUẤT BẢN TỔNG HỢP THÀNH PHỐ HỒ CHÍ MINH 2004**

# Lời nói đầu

Xin chào các bạn học sinh, sinh viên và quý độc giả yêu thích bộ môn tin học. Lời đầu tiên, chúng tôi xin chân thành cám ơn các bạn vì trong thời gian qua đã đón đọc những tập sách về ứng dụng và lập trình của chúng tôi biên soạn. Những ý kiến đóng góp và nhận xét của các bạn sinh viên và quý độc giả xa gần là điều khích lệ vô cùng to lớn và đã động viên chúng tôi tiếp tục nghiên cứu những lĩnh vực mới trong bộ môn tin học này. Hôm nay chúng tôi hoàn thành tập sách "**Bạn sẽ thành thạo C++ chỉ trong 21 ngày**" sao cho thật là ngắn gọn và dễ hiểu, có thêm phần hướng dẫn các câu lệnh và thuật toán trong các chương trình. Nghiên cứu tập sách này, chắc chắn các bạn sẽ cảm thấy hài lòng, những kiến thức hướng dẫn trong các chương trình mẫu sẽ giúp các bạn hiểu phần lý thuyết một cách thấu đáo hơn. Hy vọng tập sách này sẽ là động cơ thúc đẩy bạn yêu thích lập trình và sẽ đưa bạn tiến sâu hơn trên con đường tin học của mình. Bạn nên tìm đọc song song với cuốn này là tập sách "**Bạn sẽ thành thạo Pascal chỉ trong 21 ngày**" của cùng tác giả. Sau khi đọc xong bạn nên tìm đọc tiếp 2 tập sách "**Tự học lập trình Visual Basic 6.0 qua các chương trình mẫu**" và "**Tự học lập trình Visual C++ 6.0 qua các chương trình mẫu**" của cùng tác giả. Những tập sách này chắc chắn sẽ làm các bạn thích thú và hài lòng. Chúng tôi đã nhận được sự chúc mừng và khích lệ từ các bạn sinh viên và học sinh gửi tới. Và giờ đây xin cám ơn bạn vì đang đọc tập sách này của chúng tôi.

Mặc dầu đã có nhiều cố gắng khi biên soạn, nhưng cũng không khỏi có những thiếu sót, chúng tôi mong được sự đóng góp ý kiến của quý độc giả xa gần, để lần tái bản sau sẽ càng tốt đẹp hơn. Chúc các bạn học giỏi và đóng góp trí tuệ mình cho đất nước thân yêu của chúng ta.

***Chúng tôi thiết nghĩ, đây là một cuốn sách không thể thiếu được trong tủ sách tin học của bạn.***

Mùa hè năm 2004
Tác giả
Đậu Quang Tuấn

Mục lục

# Bài 1 *(ngày 1)*

# Cấu trúc của một chương trình C++

- **Khai báo các tập tin tiêu đề**
- **Khai báo các chỉ dẫn vĩ mô**
- **Khai báo các biến**
- **Khai báo các mảng**
- **Khai báo các hàm, các thủ tục tự tạo (*có thể để phía sau*)**

```
main ()
  {
      ...............
      ...............
      return 0;
  }
```

Đầu tiên là khai báo các tập tin tiêu đề, các tập tin này có phần mở rộng là **.h**. Khi chúng ta sử dụng các lệnh, các hàm... thuộc tập tin nào, bắt buộc ta phải khai báo, nếu không khai báo, máy sẽ không biết. Chính vì vậy chúng tôi có nhắc các bạn nhớ khai báo đường dẫn chứa các tập tin này.

*Ví dụ:*

Khi cần dùng hàm để xoá thông tin trên màn hình, ta phải khai báo tập tin tiêu đề có tên <**conio.h**>. Khi cần dùng các dòng nhập xuất cin, cout ta phải khai báo tập tin tiêu đề có tên là <**iostream.h**>. Khi dùng các hàm trong toán học như abs(), sqrt(), pow()... ta phải khai báo tập tin tiêu đề có tên <**math.h**>. Khi dùng các hàm về chuỗi như strcpy(), strlen(), strcat() ta phải khai báo tập tin tiêu đề có tên <**string.h**>... Chúng ta sẽ khai báo các tập tin tiêu đề như sau:

```
#include <iostream.h>
#include <conio.h>
```

```
#include <math.h>
#include <string.h>
```

Bạn lưu ý một điều quan trọng, trong C++, bạn phải đánh vào kí tự thường, khi bạn khai báo, ngay cả các kiểu, tên các hàm, tên các phát biểu... đều phải như vậy. Nếu dùng chữ in hoa, máy sẽ báo sai như sau:

*Ví dụ:* Bạn khai báo tập tin tiêu đề là:

```
#Include <iostream.h>
```

Nếu bạn dùng chữ I hoa, không dùng chữ i thường, khi compile chương trình bạn sẽ thấy thông báo câu "*unknown preprocessor directive: 'Include'*"

Nếu bạn dùng một hàm nào đó mà không khai báo tập tin tiêu đề chứa hàm đó, bạn sẽ được thông báo "function... tên hàm... should have a prototype"

Bạn có thể vào phần Help, vào phần mục lục, click xem phần các tập tin tiêu đề, bạn sẽ thấy tên các tập tin tiêu đề của Borland C++. Muốn biết tập tin tiêu đề nào đó chứa các hàm gì, bạn đưa vệt sáng đến tập tin tiêu đề đó, click chuột, bạn sẽ thấy tên của các hàm, C++ sẽ diễn giải các hàm đó. Có những hàm cho sẵn ví dụ, nếu giỏi Anh văn, các bạn có thể học hỏi trong phần hướng dẫn này.

Sau khi khai báo các tập tin tiêu đề, các bạn có thể khai báo các chỉ dẫn vĩ mô #define dùng để định nghĩa một phát biểu mà ta sử dụng trong hàm main().

*Ví dụ:* Bạn có một chương trình như sau:

```
#include <iostream.h>
#include <conio.h>
#define GIOITHIEU "Chuc mung ban den voi C++"
main()
{
  clrscr();
  cout<<GIOITHIEU;
  getch();
  return 0;
}
```

Nhấn Ctrl+F9 để chạy chương trình, bạn sẽ thấy trên màn hình câu "Chuc mung ban den voi C++"

Dùng các dòng xuất **cout** và dòng nhập cin cùng với các toán tử << và >>, ta phải khai báo tập tin tiêu đề <iostream.h>. Dùng hàm clrscr() và getch() ta phải khai báo tập tin tiêu đề <conio.h>.

Nếu không có hàm getch(),khi chạy chương trình, màn hình nháy lên rồi vẫn y như cũ, muốn thấy, bạn phải nhấn ALT+F5.

Nhờ các chỉ dẫn vĩ mô này, bạn sẽ thấy chương trình sáng sủa hơn.

Sau chỉ dẫn vĩ mô là khai báo các biến dùng trong chương trình. Để khai báo một biến cho chương trình, đầu tiên sẽ là kiểu của biến, kế đến là khoảng trắng, sau đó là tên biến, cuối cùng là dấu chấm phẩy.

Dòng xuất cout đi cùng với toán tử << sẽ cho kết quả trên màn hình là nội dung chứa trong cặp dấu nháy kép, và dòng nhập cin đi cùng toán tử >> sẽ cho ta nhập vào một giá trị nào đó và gán vào biến đi sau toán tử >>. Bạn chỉ mới thấy dòng xuất trong chương trình trên, bạn sẽ thấy dòng nhập trong các chương trình sau

*Ví dụ:*

```
int  x;
int  m;
float y;
char so[12];
int mang[10];
int hoc[5][5];
```

Trong trường hợp trên, bạn khai báo biến x, biến m có kiểu nguyên (*bạn lưu ý các kí tự đầu của kiểu phải là chữ thường*). Biến y có kiểu là số thực, biến so có kiểu chuỗi, độ lớn là 12, khi nhập, bạn chỉ nhập được chuỗi có độ dài là 11 mà thôi, thành phần cuối cùng sẽ là kí tự trống '\0'. Kế đến là khai báo mảng một chiều có tên là mang, độ lớn là 10. Tiếp đến là khai báo mảng hai chiều có tên Hoc[5][5].

*Ví dụ:*

```
#include <iostream.h>
#include <conio.h>
#define GIOITHIEU "Ban nhap x va y, ban se thay ket qua x+y"
int x;
int y;
main()
{
  clrscr();
  cout<<GIOITHIEU;
  cout<<"\n";
  cout<<"Ban nhap vao gia tri cua x=";
  cin>>x;
```

```
  cout<<"Ban nhap vao gia tri cua y=" ;
  cin>>y;
  cout<<"Ket qua x+y= "<<x+y;
  getch();
  return 0;
}
```

Trong chương trình trên, chúng tôi dùng 2 biến, nên chúng tôi phải khai báo đầy đủ cả hai. Chương trình trên sẽ cho bạn nhập vào giá trị của x và y, sau đó sẽ báo ra kết quả của x+y. Phần khai báo các mảng và các hàm, chúng tôi sẽ đề cập sau.

Tiếp đến là hàm main(), hàm này bắt đầu bằng dấu

```
{
    các phát biểu ...
    và kết thúc bằng return 0;
}
```

Hàm main() này phải có trong mọi chương trình của C++.

Để ghi chú một câu nào đó, bạn dùng dấu //

*Ví dụ:*

// đây là dòng ghi chú.

Để ghi chú nhiều dòng, bạn có thể dùng cặp dấu như sau:

```
/* ...................................
   ................................... */
```

Trong chương trình, những phần có từ hai lệnh trở lên, bạn phải dùng cặp móc để bao như sau:

```
{
  ...................
  ...................
}
```

Kí hiệu "\n" dùng để xuống dòng, bạn có thể dùng như trên, các bạn có thể khai báo ở đầu hoặc cuối chuỗi trong dấu nháy kép, bạn cũng có thể dùng tiếp sau chuỗi và dấu <<. Bạn sẽ thấy chúng tôi dùng trong các ví dụ sắp tới, nó có cùng ý nghĩa là xuống dòng.

## Các kiểu của biến trong C++

- Kiểu **int**: Là số nguyên (dùng hệ số thập phân), chứa trong 2 byte, nằm trong khoảng từ -32768 đến 32767.
- Kiểu **unsigned int**: Là kiểu số nguyên dương từ 0 đến 65.535.

- Kiểu **long int**: Kiểu số nguyên dài
  Từ -2.147.483.648 đến 2.147.483.647
- Kiểu **float**: Là số thực, từ $3.4x10^{-38}$ đến $3.4x10^{38}$
- Kiểu **double**: Là số thực dài, từ $1.7x10^{-308}$ đến $1.7x10^{308}$
- Kiểu **long double**: Là số thực rất dài
  Từ $3.4x10^{-4932}$ đến $1.1x10^{4932}$
- Kiểu **char**: Là kiểu kí tự đơn, từ –128 đến 127
- Kiểu **char tenbien [n]**: Là kiểu chuỗi, gồm n kí tự đơn

## Các phép toán về số học

| | |
|---|---|
| * | là phép toán nhân |
| / | là phép toán chia |
| + | là phép toán cộng |
| - | là phép toán trừ |
| % | là phép toán tìm số dư |

## Toán tử gán

- Trong C++ toán tử gán là "="

*Ví dụ:* x=2;
y=5;

## Các phép toán về luận lý

&& phép toán And
ví dụ: (x>10)&&(y<1)

| | phép toán Or
ví dụ: (x>=5)| |(y<=4)

! Phép toán phủ định

## Các phép toán so sánh

- Trong C++ các toán tử so sánh như sau:

| | |
|---|---|
| < | nhỏ hơn |
| > | lớn hơn |
| == | bằng |
| <= | nhỏ hơn hoặc bằng |
| >= | lớn hơn hoặc bằng |
| != | khác |

*Ví dụ:* x<y, m>n, a==b, j<=l, z>=o, u!=v

## Toán tử tăng giảm

Để tăng n lên một đơn vị ta viết n++ hoặc ++n, hai kí hiệu này sẽ khác biệt nhau.

*Ví dụ:*

```
#include <iostream.h>
#include <conio.h>
#define GIOITHIEU "Ban nhap x va y, ban se thay ket qua x+y"
int x;
int y;
int z;
main()
{
  clrscr();
  cout<<GIOITHIEU;
  cout<<"\n";
  cout<<"Ban nhap vao gia tri cua x=";
  cin>>x;
  cout<<"Ban nhap vao gia tri cua y=";
  cin>>y;
  cout<<"Ket qua x+y="<<x+y<<"\n";
  z=x+y;
  cout<<"\n";
  cout<<z++<<"\n";
  cout<<z;
  getch();
  return 0;
}
```

▪ Chạy thử chương trình, giả sử bạn nhập x=2, y=3, bạn sẽ thấy thông báo như sau:

```
x+y=5
5
6
```

▪ Vậy bạn thấy dòng cout<<z++ là lệnh tăng z lên một đơn vị nhưng chưa thực hiện, nên vẫn cho kết quả z=x+y=5, dòng xuất sau cout<<z sẽ cho ta kết quả là 6.

▪ Bạn thấy dòng nhập cin cùng với toán tử >> sẽ lấy giá trị 2 mà bạn nhập và cho vào biến x, giá trị 3 cho biến y. Vậy bạn đã biết công dụng của dòng nhập cin và toán tử >>. Nếu sửa lại chương trình như sau:

```
#include <iostream.h>
#include <conio.h>
#define GIOITHIEU "Ban nhap x va y, ban se thay ket qua x+y"
int x;
int y;
int z;
main()
{
  clrscr();
  cout<<GIOITHIEU;
  cout<<"\n";
  cout<<"Ban nhap vao gia tri cua x=";
  cin>>x;
  cout<<"Ban nhap vao gia tri cua y=";
  cin>>y;
  cout<<"Ket qua x+y="<<x+y<<"\n";
  z=x+y;
  cout<<"\n";
  cout<<++z<<"\n";
  cout<<z;
  getch();
  return 0;
}
```

▪ Chạy thử chương trình, giả sử bạn nhập x=2, y=3, bạn sẽ thấy thông báo như sau:

```
x+y=5
6
6
```

▪ Bạn thấy ++z sẽ tăng z lên một đơn vị và thực hiện ngay, kết quả sau thì vẫn như vậy. Bạn chú ý hai trường hợp này để áp dụng cho từng trường hợp cần thiết.

▪ Tương tự như vậy, để giảm n một đơn vị ta viết n-- hoặc –n, hai kí hiệu này sẽ khác biệt nhau:

*Ví dụ:*

```
#include <iostream.h>
#include <conio.h>
#define GIOITHIEU "Ban nhap x va y, ban se thay ket qua x+y"
int x;
int y;
int z;
main()
{
  clrscr();
  cout<<GIOITHIEU;
  cout<<"\n";
  cout<<"Ban nhap vao gia tri cua x= ";
  cin>>x;
  cout<<"Ban nhap vao gia tri cua y=";
  cin>>y;
  cout<<"Ket qua x+y="<<x+y<<"\n";
  z=x+y;
  cout<<"\n";
  cout<<z--<<"\n";
  cout<<z;
  getch();
  return 0;
}
```

• Chạy thử chương trình, giả sử bạn nhập x=4, y=5, bạn sẽ thấy thông báo như sau:

```
x+y=9
9
8
```

• Bạn thấy dòng cout<<z-- là lệnh giảm z một đơn vị nhưng chưa thực hiện, nên vẫn cho kết quả z=x+y=9, dòng xuất sau cout<<z sẽ cho ta kết quả là 8. Nếu sửa lại chương trình như sau:

```
#include <iostream.h>
#include <conio.h>
#define GIOITHIEU "Ban nhap x va y, ban se thay ket qua x+y"
int x;
int y;
int z;
main()
```

```
{
  clrscr();
  cout<<GIOITHIEU;
  cout<<"\n";
  cout<<"Ban nhap vao gia tri cua x=";
  cin>>x;
  cout<<"Ban nhap vao gia tri cua y=";
  cin>>y;
  cout<<"Ket qua x+y="<<x+y<<"\n";
  z=x+y;
  cout<<"\n";
  cout<<--z<<"\n";
  cout<<z;
  getch();
  return 0;
}
```

▪ Chạy thử chương trình, giả sử bạn nhập x=4, y=5, bạn sẽ thấy thông báo như sau:

x+y=9
8
8

▪ Bạn thấy -- z sẽ giảm ngay một đơn vị, nên bạn sẽ có giá trị là 8, giá trị sau cũng vẫn là 8.

## Kết hợp toán tử tính toán và toán tử gán

| | | |
|---|---|---|
| x=x+y | tương đương với | x+=y |
| x=x-y | tương đương với | x-=y |
| x=x*y | tương đương với | x*=y |
| x=x/y | tương đương với | x/=y |
| x=x%y | tương đương với | x%=y |

## Toán tử điều kiện

Đó là toán tử ?

Có dạng .....?.....

Ketqua=điều kiện ? biểu thức 1 : biểu thức 2

Nếu điều kiện đúng thì ketqua bằng biểu thức 1, nếu điều kiện sai thì Ketqua bằng biểu thức 2.

*Ví dụ:* Bạn nhìn chương trình sau, và đoán cho chúng tôi biến k và biến h bằng bao nhiêu, nếu như bạn nhập x=2, y=3.

```
#include <iostream.h>
#include <conio.h>
#define GIOITHIEU "Ban nhap x va y, ban se thay ket qua x+y"
int x;
int y;
int z;
int n,m,k,h;
main()
{
  clrscr();
  cout<<GIOITHIEU;
  cout<<"\n";
  cout<<"Ban nhap vao gia tri cua x=";
  cin>>x;
  cout<<"Ban nhap vao gia tri cua y=";
  cin>>y;
  cout<<"Ket qua x+y= "<<x+y<<"\n";
  z=x+y;
  m=z++;
  n=++z;
  k=(n>m) ? m:n;
  cout<<"\n";
  cout<<"k= "<<k;
  h=(n>m) ? n:m;
  cout<<"\n";
  cout<<"h= "<<h;
  getch();
  return 0;
}
```

▪ Bạn thử đoán nhé, nếu muốn biết k và h bằng bao nhiêu với giá trị của x và y như trên, bạn gõ chương trình này và cho chạy thử.

# Bài 2 *(ngày 2 và 3)*

# Các cấu trúc điều khiển trong C++

- Trong C++ có các phát biểu sau:

☞ **Phát biểu ghép {.....}**

**Dạng tổng quát**

```
{
    S1;
    S2;
    ...
    Sn;
}
```

☞ **Phát biểu IF**

*Dạng 1:*

```
        if (Điều kiện )
            s1;
```

- Phát biểu dạng 1 ở trên sẽ xét điều kiện theo sau if, nếu đúng thì thực hiện lệnh s1, nếu sai thì coi như phát biểu này đã thực hiện xong.

*Ví dụ:*

```
// Phatbieu_If_khong_else.cpp
#include <iostream.h>
#include <conio.h>
int x;
int y;
int z;
long int kq;
main()
{
 clrscr();
```

```
  cout<<"Be nhap vao gia tri cua x=";
  cin>>x;
  cout<<"\n Be nhap vao gia tri cua y=";
  cin>>y;
  cout<<"\n Be nhap vao gia tri cua z=";
  cin>>z;
  cout<<"\n";
  cout<<"Be doan va nhap vao ket qua cua x+y+z=";
  cin>>kq;
  if (kq==x+y+z)
     cout<<"Hoan ho Be da doan dung";
  if (kq!=x+y+z)
     cout<<"Rat tiec Be da doan sai";
  getch();
  return 0;
}
```

⇨ Chương trình trên cho người sử dụng nhập vào 3 giá trị của x, y và z. Sau đó người sử dụng sẽ đoán và nhập vào kết quả của x+y+z.

⇨ Nếu x+y+z bằng với kết quả nhập, sẽ thông báo là làm đúng. Ngược lại sẽ thông báo sai

⇨ Ở trên chúng tôi dùng hai phát biểu if để xét hai điều kiện riêng biệt. Và chỉ có một lệnh, nếu bạn có nhiều hơn một phát biểu, các bạn phải dùng cặp dấu móc.

*Ví dụ:*

**// Phatbieu_If_khong_else_conhieulenh.cpp**

```
#include <iostream.h>
#include <conio.h>
int x;
int y;
int z;
long int kq;
main()
{
  clrscr();
  cout<<"Be nhap vao gia tri cua x=";
  cin>>x;
  cout<<"\n Be nhap vao gia tri cua y=";
```

```
    cin>>y;
    cout<<"\n Be nhap vao gia tri cua z=";
    cin>>z;
    cout<<"\n";
    cout<<"Be doan va nhap vao ket qua cua x+y+z =";
    cin>>kq;
    if (kq==x+y+z)
     {
       cout<<"Hoan ho Be da doan dung";
       cout<<"\n";
       cout<<"Be nay se hoc toan tot";
     }
    if (kq!=x+y+z)
     {
       cout<<"Rat tiec Be da doan sai"<<"\n";
       cout<<"Be can co gang nhieu hon nua";
     }
  getch();
  return 0;
}
```

⇨ Nội dung của chương trình này cũng giống như chương trình trên, nó chỉ khác là có nhiều phát biểu hơn, nên các bạn phải dùng cặp dấu móc. Thông tin sẽ là các dòng trong hai dấu nháy kép (" ").

*Dạng 2:*

```
if (Điều kiện)
      s1;
else
      s2;
```

⇨ Phát biểu if ở dạng 2 này sẽ xét điều kiện theo sau if, nếu điều kiện đúng (True), lệnh s1 được thực hiện, nếu sai (False), lệnh s2 được thực hiện. (*Bạn ngầm hiểu s1, và s2 có thể là một hoặc nhiều phát biểu, nếu từ hai trở lên thì ta phải dùng cặp dấu móc*)

⇨ Các phát biểu If trên có thể lồng vào nhau. s1, s2 có thể là một phát biểu if khác, hoặc các vòng lặp ...

*Ví dụ:*

```
// Phatbieu_If_co_else.cpp
```

```
#include <iostream.h>
#include <conio.h>
int x;
int y;
int z;
long int kq;
main()
{
  clrscr();
  cout<<"Be nhap vao gia tri cua x=";
  cin>>x;
  cout<<"\n Be nhap vao gia tri cua y=";
  cin>>y;
  cout<<"\n Be nhap vao gia tri cua z=";
  cin>>z;
  cout<<"\n";
  cout<<"Be doan va nhap vao ket qua cua x+y+z=";
  cin>>kq;
  if (kq==x+y+z)
      cout<<"Hoan ho Be da doan dung";
  else
      cout<<"Rat tiec Be da doan sai";
  getch();
  return 0;
}
```

⇨ Trong chương trình này bạn không dùng hai phát biểu if không có else như ở trên, bạn chỉ cần dùng một phát biểu if có else là có thể thay thế được và sẽ được kết quả như bài trước. Bạn cũng có thể dùng phát biểu if... else lồng nhau, bạn có thể xem ví dụ sau:

**// Phatbieu_If_co_else_liennhau .cpp**

```
#include <iostream.h>
#include <conio.h>
int Toan;
int Ly;
int Hoa;
int kq;
main()
```

```
{
  clrscr();
  cout<<"Ban nhap vao diem Toan (toi da la 10):";
  cin>>Toan;
  cout<<"\n Ban nhap vao diem Ly (toi da la 10):";
  cin>>Ly;
  cout<<"\n Ban nhap vao diem Hoa (toi da la 10):";
  cin>>Hoa;
  kq=Toan+Ly+Hoa;
  if (kq>=25)
   {
     cout<<"Hoc sinh nay duoc vao dai hoc";
     cout<<"\n Khong can phai xem diem khong che";
   }
  else
   {
     if ((kq>=2)&&(Toan>=7)&&(Ly>=7)&&(Hoa>=7))
           cout<<"Hoc sinh nay dau, vi cac mon deu tu 7 tro len";
     else
           cout<<"Khong du tieu chuan ve diem de vao dai hoc";
   }
 getch();
 return 0;
}
```

⇨ Trong chương trình trên, bạn thấy chúng tôi dùng phát biểu if ... else lồng nhau. Nếu tổng số điểm của 3 môn mà lớn hơn hay bằng 25 thì sẽ báo là đậu vào đại học. Ngược lại, xét tiếp điều kiện, nếu tổng số điểm của 3 môn chỉ cần lớn hơn hoặc bằng 21 và mỗi môn phải lớn hơn hoặc bằng 7 thì cũng đậu. Các trường hợp còn lại sẽ không đủ tiêu chuẩn.

*Ví dụ:*

◎ Chạy thử chương trình, giả sử nhập vào điểm toán, lý và hoá đều bằng 7. Bạn sẽ thấy báo là trúng tuyển.

◎ Nếu bạn nhập toán và lý bằng 10, hoá bằng 5, bạn cũng sẽ thấy thông báo là đậu.

◎ Cả hai trường hợp trên đều thực hiện dòng lệnh trong if đầu tiên.

◎ Bây giờ bạn thử nhập môn toán, lý và hoá đều bằng 7, bạn sẽ thấy thông báo nội dung đậu trong if sau else.

◎ Nếu bạn nhập toán và lý bằng 10, hoá là 4, bạn sẽ thấy thông báo là không trúng tuyển.

◎ Nếu bạn nhập toán bằng 8, lý bằng 8 và hoá bằng 6, bạn cũng thấy thông báo là không trúng tuyển, mặc dầu tổng số điểm lớn hơn 21, nhưng vì có điểm hoá nhỏ hơn 7 nên không thoả ...

◎ Bạn có thể nhập thêm một số trường hợp nữa để kiểm tra.

☞ **Phát biểu SWITCH**

```
switch ( biểu thức )
  {
    case  giá trị 1 : s1; break;
    case  giá trị 2 : s2; break;
    ..................
    ..................
    case giá trị n : sn; break;
    default : sm;
  }
```

• Đầu tiên các giá trị được xem xét và đối chiếu với trị của biểu thức, nếu trị đó nằm trong case nào, thì câu lệnh (*hoặc nhiều lệnh*) sau case đó sẽ được thực hiện, sau đó gặp lệnh break sẽ thoát khỏi phát biểu switch. Nó sẽ kiểm tra từ trên xuống, nếu không có giá trị của case nào thoả, nó sẽ thực hiện câu lệnh (*hoặc nhiều lệnh*) theo sau default, và kết thúc, nếu không có default, nó sẽ kết thúc và không làm gì cả (vì không có giá trị nào sau case thoả).

*Nhận xét:*

• Phát biểu switch là một dạng làm gọn của phát biểu if. Mọi phát biểu switch đều có thể viết lại thành phát biểu if. Bạn nên dùng switch khi lựa chọn nhiều trường hợp, ví dụ như khi xếp loại học lực của học sinh... Bạn có thể xem ví dụ sau:

```
// Phatbieu_switch_khong_default.cpp
#include <iostream.h>
#include <conio.h>
```

```
int x;
main()
{
  clrscr();
  cout<<"Be nhap vao  so cua mot THANG";
  cin>>x;
  switch (x)
    {
        case 1: cout<<"Day la thang GIENG"; break;
        case 2: cout<<"Day la thang HAI"; break;
        case 3: cout<<"Day la thang BA"; break;
        case 4: cout<<"Day la thang TU"; break;
        case 5: cout<<"Day la thang NAM"; break;
        case 6: cout<<"Day la thang SAU"; break;
        case 7: cout<<"Day la thang BAY"; break;
        case 8: cout<<"Day la thang TAM"; break;
        case 9: cout<<"Day la thang CHIN"; break;
        case 10: cout<<"Day la thang MUOI"; break;
        case 11: cout<<"Day la thang MUOI MOT"; break;
        case 12: cout<<"Day la thang MUOI HAI"; break;
    }
  getch();
  return 0;
}
```

- Trong chương trình này, chúng tôi khai báo biến x có kiểu nguyên. Trong phát biểu switch, chúng tôi dùng 12 giá trị từ 1 đến 12 để báo ra các tháng bằng các giá trị tương ứng của số nhập vào. Nếu bạn nhập vào số 12, chương trình sẽ báo đây là tháng12. Sau đó thoát khỏi phát biểu. Nếu người sử dụng nhập vào một số nguyên không nằm trong các giá trị được khai báo sau case, nó sẽ không làm gì hết trong phát biểu switch, nó tiếp tục thực hiện các lệnh theo sau phát biểu này. Ví dụ bạn nhập vào số 13 chẳng hạn, chương trình sẽ không báo ra điều gì cả, nếu trong phát biểu switch, bạn thêm một dòng:

```
case 13: cout<<"Day la thang thu MUOI BA"; break;
```

◎ Chương trình sẽ báo ra "Day la thang thu MUOI BA".

◎ Để hay hơn, khi người sử dụng nhập một số nguyên không nằm trong khoảng từ 1 đến 12, chương trình phải nhắc nhở một điều gì đó, bạn phải dùng thêm default:

```
// Phatbieu_switch_co_default.cpp
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  cout<<"Be nhap vao  so cua mot THANG";
  cin>>x;
  switch (x)
    {
        case 1: cout<<"Day la thang GIENG"; break;
        case 2: cout<<"Day la thang HAI"; break;
        case 3: cout<<"Day la thang BA"; break;
        case 4: cout<<"Day la thang TU"; break;
        case 5: cout<<"Day la thang NAM"; break;
        case 6: cout<<"Day la thang SAU"; break;
        case 7: cout<<"Day la thang BAY"; break;
        case 8: cout<<"Day la thang TAM"; break;
        case 9: cout<<"Day la thang CHIN"; break;
        case 10: cout<<"Day la thang MUOI"; break;
        case 11: cout<<"Day la thang MUOI MOT"; break;
        case 12: cout<<"Day la thang MUOI HAI"; break;
        default:
         {
           cout<<"Khong co thang nay.";
           cout<<"Be khong biet mot chut gi ve ngay thang";
         }
    }
  getch();
  return 0;
}
```

⇨ Bài này cũng giống như bài trên, nhưng khi người sử dụng nhập một số không phải từ 1 đến 12, các dòng lệnh sau default: sẽ được thực

hiện. Sau các câu lệnh này, các bạn thấy không có lệnh break, bạn không cần dùng vì đây là phát biểu sau cùng. Nếu đi đến đây, điều ấy chứng tỏ rằng các giá trị sau case không có giá trị nào cả.

⇨ Một câu hỏi được đặt ra là, sau case có thể được dùng nhiều giá trị hay không, bạn có thể xem trả lời bằng ví dụ sau đây

```
// switch_dungnhieu_giatri_saucase.cpp
// khong dung duoc
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  cout<<"Be nhap vao so cua mot THANG";
  cin>>x;
  switch (x)
    {
      case 1,3,5,7,8,10,12:cout<<"Thang:"<<x<<"co 31 ngay"; break;
      case 4,6,9,11: cout<<"Thang:"<<x<<"co 30 ngay"; break;
      case 2: cout<<"Thang:"<<x<<"co 28 ngay"; break;
      default:
          {
              cout<<"Khong co thang nay.";
              cout<<"Be khong biet mot chut gi ve ngay thang";
          }
    }
  getch();
  return 0;
}
```

⇨ Nếu bạn viết một chương trình như trên, khi biên dịch không báo sai, các giá trị sau case, trình biên dịch chỉ nhận ra các số đứng liền trước dấu hai chấm ":"

⇨ Khi chạy thử chương trình, nếu bạn nhập các số 12, 11 hoặc 2, chương trình sẽ báo ra câu lệnh theo sau giá trị đó. Nhưng nếu bạn nhập các số ngoài 3 số này, máy sẽ báo không có tháng này.

⇨ Bạn không được dùng các giá trị như trên. Tuy nhiên trong ngôn ngữ PASCAL thì bạn có thể dùng được dạng trên.

☞ **Vòng lặp WHILE**

**while (Điều kiện)**

**s1;**

- Dùng để lặp đi lặp lại một công việc gì đó khi điều kiện còn đúng.
- Trước hết điều kiện được xét, nếu giá trị là False, vòng lặp while kết thúc, nếu True thì phát biểu s1 được thực hiện. Sau khi thực hiện xong, quay lại kiểm tra Điều kiện, và công việc cứ tiếp tục như lập luận ở trên. Bạn có thể tham khảo ví dụ sau:

*Ví dụ:*

```
// Vonglap_while.cpp
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  x=1;
  while (x<=12)
    {
        cout<<"Day la thang:  " << x << "\n";
        x=x+1;
    }
    getch();
    return 0;
}
```

◎ Khi bạn chạy chương trình này, biến x=1, vòng lặp while sẽ thực hiện khi x nhỏ hơn hoặc bằng 12. Khi x=13 nó sẽ thoát khỏi vòng lặp while.

◎ Khi dùng vòng lặp while, giá trị của biến x không được tăng tự động, bạn phải dùng câu lệnh x=x+1; để tăng biến x lên một đơn vị sau mỗi lần lặp.

◎ Chạy thử chương trình, bạn sẽ có kết quả trên màn hình như sau:

Đây là tháng : 1

```
Đây là tháng : 2
Đây là tháng : 3
Đây là tháng : 4
Đây là tháng : 5
Đây là tháng : 6
Đây là tháng : 7
Đây là tháng : 8
Đây là tháng : 9
Đây là tháng : 10
Đây là tháng : 11
Đây là tháng : 12
```

◎ Bạn có thể xem tiếp thêm một ví dụ về vòng lặp while như sau

```
// Vonglap_while.cpp
#include <iostream.h>
#include <conio.h>
#include <string.h>
int i;
char chuoi[20];
main()
{
  clrscr();
  cout<<"Ban nhap vao mot chuoi co chieu dai nho hon 20:";
  cin>>chuoi;
  i=0;
  while (i<strlen(chuoi))

    {
        cout<<"Cac ki tu trong chuoi the hien theo chieu doc:"<<chuoi[i]<<"\n";
        i=i+1;
    }
    getch();
    return 0;
}
```

⇨ Bạn nhập vào một chuỗi là một câu nào đó, chuỗi được xem như là một mảng. Muốn truy xuất một phần tử nào đó, bạn dùng tên của chuỗi và vị trí cần truy xuất. Bạn nhớ chỉ số chạy trong chuỗi bắt đầu từ 0, trừ khi

bạn áp đặt khi nhập. Bạn sẽ thấy khi chúng tôi dùng trong các chương trình sau này.

⇨ Vòng lặp while này sẽ thực hiện khi biến i còn nhỏ hơn chiều dài của chuỗi.

⇨ Dùng biến i xác định vị trí chạy của chuỗi để lấy ra các phần tử tương ứng của chuỗi.

⇨ Phải tăng biến i lên một đơn vị sau một lần lặp.

⇨ Bạn chạy chương trình, giả sử nhập câu LAPTRINHC++, bạn sẽ thấy kết quả như sau:

Các kí tự trong chuỗi thể hiện theo chiều dọc : L
Các kí tự trong chuỗi thể hiện theo chiều dọc : A
Các kí tự trong chuỗi thể hiện theo chiều dọc : P
Các kí tự trong chuỗi thể hiện theo chiều dọc : T
Các kí tự trong chuỗi thể hiện theo chiều dọc : R
Các kí tự trong chuỗi thể hiện theo chiều dọc : I
Các kí tự trong chuỗi thể hiện theo chiều dọc : N
Các kí tự trong chuỗi thể hiện theo chiều dọc : H
Các kí tự trong chuỗi thể hiện theo chiều dọc : C
Các kí tự trong chuỗi thể hiện theo chiều dọc : +
Các kí tự trong chuỗi thể hiện theo chiều dọc : +

☞ **Vòng lặp DO ... WHILE**

```
do
s1;
while  ( Điều kiện );
```

▪ Dùng để lặp đi lặp lại một công việc nào đó khi điều kiện đúng trong while còn đúng, và kết thúc khi điều kiện này sai.

▪ Đầu tiên các phát biểu trong thân của phát biểu do ... while sẽ được thực hiện, sau đó mới kiểm tra điều kiện, nếu đúng thì vòng lặp tiếp tục, nếu sai, sẽ kết thúc vòng lặp.

Nhận xét:

▪ While và do ... while đều là những vòng lặp. Đối với while thì điều kiện được xét trước, nếu đúng thì các phát biểu trong thân mới được thực hiện. Đối với do ... while thì các phát biểu được thực hiện trước, sau đó mới xét đến điều kiện. Vậy đối với do ... while thì dù điều kiện đúng hay

sai, các lệnh trong thân sẽ thực hiện ít nhất là một lần, còn while thì khác. Do đó tùy theo chương trình, bạn phải cân nhắc để dùng cho đúng. Bạn có thể theo dõi một ví dụ sau:

```
// Vonglap_do_while.cpp
#include <iostream.h>
#include <conio.h>
int i;
main()
{
  clrscr();
  i=1;
  do
    {
        cout << "Day la thang:  "<<i<<"\n";
        i=i+1;
    }
  while (i!=13);
  cout<<"Ban thay mot nam co 12 thang nhu tren";
  getch();
  return 0;
}
```

- Khi chạy chương trình trên, nó sẽ thực hiện khi i khác 13, nó sẽ kết thúc khi i=13. Kết quả của chương trình trên:

Đây là tháng : 1
Đây là tháng : 2
Đây là tháng : 3
Đây là tháng : 4
Đây là tháng : 5
Đây là tháng : 6
Đây là tháng : 7
Đây là tháng : 8
Đây là tháng : 9
Đây là tháng : 10
Đây là tháng : 11
Đây là tháng : 12

- Bạn thấy một năm có 12 tháng như trên.

### ☞ Vòng lặp FOR

Khi biết trước số lần lặp, bạn dùng vòng lặp FOR

**for (giá trị đầu; điều kiện;thay đổi)**

**s1;**

- Bạn có thể tham khảo ví dụ sau đây:

```
// Vonglap_for_mot.cpp
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  for (x=1; x<=12; x++)
        cout<<"Day la thang :"<<x<<"\n";
    getch();
    return 0;
}
```

⇨ Trong chương trình này, bạn sẽ thấy vòng lặp for sẽ thực hiện bắt đầu i=1, cho đến khi i=12, sau mỗi lần tăng biến i lên một đơn vị nhờ i++.

⇨ Trong for do có i++ có sẵn trong phát biểu, nên khác với vòng lặp while. Chạy thử chương trình trên, bạn sẽ có kết quả như sau:

```
Đây là tháng : 1
Đây là tháng : 2
Đây là tháng : 3
Đây là tháng : 4
Đây là tháng : 5
Đây là tháng : 6
Đây là tháng : 7
Đây là tháng : 8
Đây là tháng : 9
Đây là tháng : 10
Đây là tháng : 11
Đây là tháng : 12
```

Nếu bạn thay đổi chương trình lại như sau:

```
// Vonglap_for_hai.cpp
```

```
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  for (x=12; x>=1; x--)
        cout<<"Day la thang:  "<<x<<"\n";
    getch();
    return 0;
}
```

- Trong chương trình này, chúng tôi cho biến i bắt đầu từ 12 chạy giảm dần đến khi i=1, sau mỗi lần giảm biến i một đơn vị nhờ câu lệnh i -- . Chạy thử chương trình trên, bạn sẽ có kết quả như sau:

Đây là tháng : 12
Đây là tháng : 11
Đây là tháng : 10
Đây là tháng : 9
Đây là tháng : 8
Đây là tháng : 7
Đây là tháng : 6
Đây là tháng : 5
Đây là tháng : 4
Đây là tháng : 3
Đây là tháng : 2
Đây là tháng : 1

### ☞ Phát biểu GOTO

- Dùng để chuyển sự thực thi của chương trình đến một phát biểu nào đó. Dạng phát biểu goto

**goto nhãn;**

*Ví dụ:*

```
Char  Hihi;
      If  (x==10)
         goto  Hihi;
         ..............
         ..............
```

```
        Hihi:
        ..................
```

Bạn xem ví dụ sau:

```
// Phatbieu_goto .cpp
#include <iostream.h>
#include <conio.h>
char Hihi;
int Toan;
int Ly;
int Hoa;
int kq;
main()
{
  clrscr();
  cout<<"Ban nhap vao diem Toan (toi da la 10):";
  cin>>Toan;
  cout<<"\n Ban nhap vao diem Ly (toi da la 10):";
  cin>>Ly;
  cout<<"\n Ban nhap vao diem Hoa (toi da la 10):";
  cin>>Hoa;
  kq=Toan+Ly+Hoa;
  if (kq==0)
     goto Hihi;
  if (kq>=25)
    {
      cout<<"Hoc sinh nay duoc vao dai hoc";
      cout<<"\n Khong can phai xem diem khong che";
    }
  else
    {
      if ((kq>=21)&&(Toan>=7)&&(Ly>=7)&&(Hoa>=7))
            cout<<"Hoc sinh nay dau, vi cac mon deu tu 7 tro len";
      else
            cout<<"Khong du tieu chuan ve diem de vao dai hoc";
    }
 Hihi:;
 getch();
 return 0;
```

}

⇨ Trong chương trình trên, chúng tôi khai báo biến Hihi có kiểu char, khi tổng số điểm của 3 môn là zero, sẽ chạy đến nhãn có tên là Hihi, bạn dùng câu lệnh Hihi:;

⇨ Trong trường hợp như trên, chương trình sẽ không báo ra gì cả, vì nhãn Hihi nằm ở cuối chương trình.

### ☞ Phát biểu BREAK

▪ Phát biểu này dùng để ngưng các vòng lặp (*for, while, do ... while*) mặc dù điều kiện của các vòng lặp vẫn còn đúng, để chuyển sang phát biểu kế tiếp hoặc để ngưng phát biểu switch như chúng tôi giới thiệu ở trên.

*Ví dụ:*

```
// Vonglap_for_hai_dung_break.cpp
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  for (x=12; x>=1; x--)
     {
         cout << "Day la thang :"<<x<<"\n";
          if (x==5)
               break;
     }
    getch();
    return 0;
}
```

▪ Chạy chương trình này ta được kết quả như sau:

```
Đây là tháng : 12
Đây là tháng : 11
Đây là tháng : 10
Đây là tháng : 9
Đây là tháng : 8
Đây là tháng : 7
Đây là tháng : 6
```

Đây là tháng : 5

- Vậy khi i=5, câu lệnh cout<< vẫn xuất ra câu: Đây là tháng 5
- Sau đó gặp phát biểu if (i==5) thoả điều kiện nên thực hiện lệnh break và thoát khỏi vòng lặp, không thực hiện tiếp cho đến khi i=1.
- Nếu bạn di chuyển lệnh cout<< ra sau lệnh break, khi i=5 thoả điều kiện, nó sẽ không thực hiện được lệnh cout<< với giá trị i=5. Do đó chương trình sẽ in ra kết quả khác, bạn sẽ thấy chương trình và kết quả khi chạy như sau:

```
// Vonglap_for_hai_dung_break.cpp
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  for (x=12; x>=1; x--)
     {
        if (x==5)
              break;
              cout<<"Day la thang:"<<x<<"\n";
     }
    getch();
    return 0;
}
```

```
Đây là tháng : 12
Đây là tháng : 11
Đây là tháng : 10
Đây là tháng : 9
Đây là tháng : 8
Đây là tháng : 7
Đây là tháng : 6
```

### ☞ Phát biểu CONTINUE

- Bạn hãy viết và chạy hai chương trình sau để biết công dụng của phát biểu continue.

```
// Vonglap_for_hai_dung_continue.cpp
```

```
#include <iostream.h>
#include <conio.h>
int x;
main()
{
  clrscr();
  for (x=12; x>=1; x--)
     {
        if (x==5)
              continue;
              cout<<"Day la thang :"<<x<<"\n";
     }
   getch();
   return 0;
}
```

- Chạy chương trình trên, bạn sẽ có kết quả như sau:

```
Đây là tháng : 12
Đây là tháng : 11
Đây là tháng : 10
Đây là tháng : 9
Đây là tháng : 8
Đây là tháng : 7
Đây là tháng : 6
Đây là tháng : 4
Đây là tháng : 3
Đây là tháng : 2
Đây là tháng : 1
```

Với vị trí của phát biểu continue như trên, ta thấy chương trình sẽ thực hiện các trường hợp của biến i từ 12 giảm dần đến 1, bỏ qua trường hợp khi i=5.

Nếu thay đổi vị trí của lệnh continue như sau:

```
// Vonglap_for_hai_dung_continue.cpp
#include <iostream.h>
#include <conio.h>
int x;
main()
{
```

```
    clrscr();
    for (x=12; x>=1; x--)
        {
            if (x==5)
                cout<<"Day la thang :"<<x<<"\n";
                continue;
        }
      getch();
      return 0;
}
```

- Chạy chương trình trên, bạn sẽ thấy kết quả như sau:

  Đây là tháng 5

- Bạn tự tìm hiểu ý nghĩa của phát biểu continue nhé.

# Bài 3 *(ngày 4)*

# Mảng trong C++

- Mảng là một phần của bộ nhớ dùng để chứa một tập hợp có thứ tự các phần tử có cùng một kiểu nào đó.

## ① Mảng một chiều

- Cách khai báo, bạn có thể khai báo theo những cách như sau :

**Kiểucủamảng tênmảng[độlớn];**

- Giữa kiểu của mảng và tên mảng có khoảng trắng.

*Ví dụ:*

Int Mang[100];
Fioat So[10];
Các thành phần của các mảng trên sẽ là:
Mang[0], Mang[1], ... , Mang[98], Mang[99].
So[0], So[1], ... , So[8], So[9].

- Bạn có thể gán các giá trị cho mảng, hoặc nhập giá trị cho mảng qua vòng lặp for, đồng thời các bạn có thể truy xuất các phần tử của mảng theo chỉ số.

*Ví dụ:*

```
Mang[0] = 250;
for (j=0; j<=99, j++)
      cin>>Mang[j]
for (j=0; j<=99, j++)
      cout<<Mang[j] << " ";
```

*Ví dụ:*

**// Mang_motchieu.cpp**

```
#include <iostream.h>
#include <conio.h>
int j;
int So[5];
main()
{
```

```
  clrscr();
  // Nhap cac phan tu vao mang So
  for (j=1; j<=5; j++)
   {
     cout<<"Ban nhap phan tu So["<<j<< "]=";
     cin>>So[j];
   }
  // Xuat cac phan tu cua mang So ra man hinh
    for (j=1; j<=5; j++)
        cout<<So[j] << " ";
    getch();
    return 0;
}
```

⇨ Trong chương trình trên, chúng tôi khai báo mảng một chiều có tên là So, độ lớn là 5, có kiểu nguyên là int.

⇨ Đầu tiên chúng tôi nhập giá trị cho các phần tử của mảng qua vòng lặp for có j chạy từ 1 đến 5 để nhập vào 5 phần tử. Dùng dòng nhập cin cùng với toán tử >>, gán cho các phần tử của mảng qua So[j] (j chạy từ 1 đến 5).

⇨ Xuất 5 phần tử của mảng ra màn hình nhờ vòng lặp for cũng chạy từ 1 đến 5, và dòng xuất cout cùng với toán tử <<.

⇨ Bạn chạy thử chương trình, giả sử bạn nhập vào các phần tử của mảng như sau:

Bạn nhập phần tử số [1]=8
Bạn nhập phần tử số [2]=23
Bạn nhập phần tử số [3]=13
Bạn nhập phần tử số [4]=25
Bạn nhập phần tử số [5]=40
Bạn sẽ có kết quả như sau:
8 23 13 25 40

### ② Mảng hai chiều

- Cách khai báo, bạn có thể khai báo theo những cách như sau:

**Kiểucủamảng tênmảng[độlớn][độlớn];**

*Ví dụ:*

```
int Mang[100][100];
```

```
Float  So[10][10];
```

- Tương tự như ở mảng một chiều, ở mảng hai chiều, các bạn cũng có thể gán, truy xuất các phần tử của mảng. Bạn có thể tham khảo trong ví dụ sau:

```
// Mang_haichieu.cpp
#include <iostream.h>
#include <conio.h>
int i,j;
int So[4][4];
main()
{
  clrscr();
  // Nhap cac phan tu vao mang So
  for (j=1; j<=4; j++)
     for (i=1; i<=4; i++)
      {
         cout<<"Ban nhap phan tu So["<<j<< "]["<<i<<"]=";
         cin>>So[j][i];
      }
// Xuat cac phan tu cua mang So ra man hinh
   for (j=1; j<=4; j++)
      for (i=1; i<=4; i++)
            cout<<So[j][i]<<"  ";
   getch();
   return 0;
}
```

⇨ Trong chương trình này, chúng tôi giới thiệu đến các bạn mảng hai chiều, chúng tôi khai báo mảng hai chiều có tên So[4][4]. Vậy mảng này sẽ có 16 phần tử, phần tử đầu tiên sẽ là So[1][1], So[1][2], ... , So[4][3], So[4][4]. Bạn sẽ thấy khi chạy thử chương trình.

⇨ Vì là mảng hai chiều, khi nhập các phần tử của mảng, các bạn phải dùng hai vòng for lồng nhau. Nó thực hiện như sau:

⇨ Đầu tiên biến j=1, các biến i của vòng for bên trong sẽ chạy từ 1 đến 4. Sau đó biến j tăng lên một tức là j=2, các biến i của vòng for bên trong sẽ chạy từ 1 đến 4, bây giờ j=3 ... quá trình tiếp tục cho đến khi j=4 và i=4. Chúng ta sẽ được 16 phần tử.

⇨ Sau khi nhập xong 16 phần tử của mảng hai chiều, chương trình sẽ xuất ra màn hình 16 phần tử của mảng.

⇨ Việc nhập và xuất chỉ khác là dùng cin và cout.

⇨ Bây giờ chúng ta chạy thử chương trình, giả sử bạn nhập vào 16 phần tử của mảng như sau:

Bạn nhập vào phần tử So[1][1]=1
Bạn nhập vào phần tử So[1][2]=1
Bạn nhập vào phần tử So[1][3]=1
Bạn nhập vào phần tử So[1][4]=1
Bạn nhập vào phần tử So[2][1]=1
Bạn nhập vào phần tử So[2][2]=1
Bạn nhập vào phần tử So[2][3]=1
Bạn nhập vào phần tử So[2][4]=1
Bạn nhập vào phần tử So[3][1]=1
Bạn nhập vào phần tử So[3][2]=1
Bạn nhập vào phần tử So[3][3]=1
Bạn nhập vào phần tử So[3][4]=1
Bạn nhập vào phần tử So[4][1]=1
Bạn nhập vào phần tử So[4][2]=1
Bạn nhập vào phần tử So[4][3]=1
Bạn nhập vào phần tử So[4][4]=1

Bạn sẽ có kết quả như sau:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

▪ Bạn thấy trong hai vòng lặp for lồng nhau, chúng tôi cho biến j và i chạy nên trong mảng nhập, các chỉ số của các phần tử sẽ thể hiện tự động như trên.

# Bài 4 *(ngày 5)*

# Chuỗi trong C++

▪ Chuỗi là một mảng mà các thành phần là các kí tự, các phần tử cuối của mảng là kí tự "\0", tức là kí tự trống, chuỗi được viết liên tiếp các kí tự giữa hai dấu nháy kép (" "), không được có kí tự trống, ta thường gọi đó là hằng chuỗi.

*Ví dụ:* Khi cần nhập chuỗi ch[5] chẳng hạn, ta chỉ có thể nhập được 4 kí tự và không được nhập kí tự trống vì như thế trình biên dịch xem như đã chấm dứt việc nhập các kí tự.

▪ Khi dùng chuỗi, người ta thường dùng các hàm thuộc về chuỗi, khi dùng các hàm này, các bạn phải khai báo tập tin tiêu đề có tên là <string.h>. Bạn khai báo chuỗi như sau:

**Char tên[độlớn];**

*Ví dụ:* Char chuoi[10];

▪ Bạn có thể xem ví dụ sau:

```
// Chuoi.cpp
#include <iostream.h>
#include <conio.h>
#include <string.h>
int j;
char chuoi[30];
main()
{
 clrscr();
 // Nhap cac phan tu vao mang chuoi
 cout<<"Ban nhap vao chuoi";
 cin>>chuoi;
 // Xuat cac phan tu cua chuoi theo chieu doc
 for (j=0; j<=strlen(chuoi)-1; j++)
   cout<<"Ki tu thu  "<<j<<"  "<<chuoi[j]<<"\n";
 getch();
 return 0;
```

}

Bạn chạy thử chương trình, giả sử bạn nhập vào tên LAPTRINHC++, bạn sẽ có kết quả như sau:

Kí tự thứ 0 L
Kí tự thứ 1 A
Kí tự thứ 2 P
Kí tự thứ 3 T
Kí tự thứ 4 R
Kí tự thứ 5 I
Kí tự thứ 6 N
Kí tự thứ 7 H
Kí tự thứ 8 C
Kí tự thứ 9 +
Kí tự thứ 10 +

- Chạy lại chương trình, sau đó nhập vào: LAPTRINH C++
- Bạn nhập có khoảng trắng giữa chữ H và C như trên, lúc này sẽ có kết quả như sau:

Kí tự thứ 0 L
Kí tự thứ 1 A
Kí tự thứ 2 P
Kí tự thứ 3 T
Kí tự thứ 4 R
Kí tự thứ 5 I
Kí tự thứ 6 N
Kí tự thứ 7 H

- Bạn thấy chương trình chỉ in ra chữ LAPTRINH, không in ra chữ C++, điều này chúng tôi đã đề cập ở trên.

# Bài 5 *(ngày 6)*

# Liệt kê

▪ Liệt kê giống như một mảng gồm các thành phần là chuỗi, đặc biệt các thành phần trong liệt kê có thể gán một giá trị nguyên nào đó.

*Khai báo:*

**enum ten {thanhphan1,thanhphan2,...,thanhphann}**

▪ Nếu bạn khai báo như vậy thì C++ sẽ gán giá trị 0 cho thanhphan1, giá trị 1 cho thanhphan2, ... , giá trị n-1 cho thanhphann.

▪ Bạn có thể xem ví dụ sau để hiểu rõ về liệt kê. Đây là một chương trình in ra 12 tháng trong một năm.

**// Kieu_Liet_Ke.cpp**

```
#include<iostream.h>
#include<conio.h>
enum cacthang {khongco,mot,hai,ba,bon,nam,sau,bay,tam,chin,muoi,muoimot,
muoihai};
int thang;
main()
 {
   clrscr();
   cout<<"Cac thang trong mot nam"<<'\n';
   for (thang=mot; thang<=muoihai;thang++)
        cout<<"Day la thang: "<<thang<<'\n';
   getch();
   return 0;
 }
```

⇨ Đầu tiên bạn khai báo kiểu liệt kê có tên Cacthang với các thành phần như trên.

⇨ Khai báo biến thang co kiểu nguyên dùng làm biến chạy trong vòng lặp For.

⇨ Vòng lặp For như trên sẽ chạy từ thang=một cho đến thang=muoihai, tương ứng là chạy từ thang=1 đến thang=12. Chúng tôi cho

biến tháng đầu tiên bằng 1 thì mới thực hiện đúng, nếu cho biến thang=khongco thì nó sẽ chạy từ 0 đến 12, không có tháng 0.

⇨ Nếu chạy thử chương trình, bạn sẽ có kết quả như sau:

Các tháng trong một năm
Đây là tháng : 1
Đây là tháng : 2
Đây là tháng : 3
Đây là tháng : 4
Đây là tháng : 5
Đây là tháng : 6
Đây là tháng : 7
Đây là tháng : 8
Đây là tháng : 9
Đây là tháng : 10
Đây là tháng : 11
Đây là tháng : 12

- Nếu bạn thay vòng lặp for trên là

```
for (thang=khongco; thang<=muoihai;thang++)
```

- Thì bạn sẽ có kết quả như sau:

Các tháng trong một năm
Đây là tháng : 0
Đây là tháng : 1
Đây là tháng : 2
Đây là tháng : 3
Đây là tháng : 4
Đây là tháng : 5
Đây là tháng : 6
Đây là tháng : 7
Đây là tháng : 8
Đây là tháng : 9
Đây là tháng : 10
Đây là tháng : 11
Đây là tháng : 12

- Nếu viết chương trình như sau:

**// Kieu_Liet_Ke.cpp**

```
#include<iostream.h>
#include<conio.h>
enum cacthang {mot, hai, ba, bon, nam, sau, bay, tam, chin, muoi, muoimot,
muoihai};
int thang;
main()
 {
   clrscr();
   cout<<"Cac thang trong mot nam"<<'\n';
   for (thang=mot; thang<=muoihai; thang++)
         cout<<"Day la thang: "<<thang+1<<'\n';
   getch();
   return 0;
 }
```

⇨ Trong chương trình này bạn khai báo kiểu liệt kê từ mot, hai, ..., muoihai. Trong trường hợp này C++ sẽ gán giá trị 0 cho mot ... và 11 cho muoihai.

⇨ Để thông báo đúng, bạn phải tăng biến thang lên 1 đơn vị trong các lần của vòng lặp for. Nên nhìn vào chương trình, bạn thấy chúng tôi đã dùng như vậy. Chạy thử chương trình, bạn sẽ thấy kết quả

```
Các tháng trong một năm
Đây là tháng : 1
Đây là tháng : 2
Đây là tháng : 3
Đây là tháng : 4
Đây là tháng :.5
Đây là tháng : 6
Đây là tháng : 7
Đây là tháng : 8
Đây là tháng : 9
Đây là tháng : 10
Đây là tháng : 11
Đây là tháng : 12
```

Để không phải tăng biến thang lên 1 đơn vị, bạn xem chương trình sau:

**// Kieu_Liet_Ke.cpp**

```
#include<iostream.h>
#include<conio.h>
enum cacthang
{mot=1,hai,ba,bon,nam,sau,bay,tam,chin,muoi,muoimot,muoihai};
int thang;
main()
 {
   clrscr();
   cout<<"Cac thang trong mot nam"<<'\n';
   for (thang=mot; thang<=muoihai;thang++)
         cout<<"Day la thang : "<< thang << '\n';
   getch();
   return 0;
 }
```

⇨ Trong chương trình trên, bạn thấy chúng tôi gán thành phần mot = 1. Chúng tôi đã đề cập ở trên, có nghĩa là bạn có thể gán các thành phần của liệt kê cho một số nguyên nào đó, thành phần kế tiếp sẽ tăng lên 1 so với số nguyên mà bạn gán.

⇨ Do đó bạn vẫn có kết quả đúng, như các ví dụ trên. Nếu không tin, bạn chạy thử chương trình để thấy.

Bạn cũng có thể thực hiện chương trình như sau:

**// Kieu_Liet_Ke.cpp**

```
#include<iostream.h>
#include<conio.h>
enum cacthang
{khongco=1,mot,hai,ba,bon,nam,sau,bay,tam,chin,muoi,muoimot,muoihai};
int thang;
main()
 {
   clrscr();
   cout<<"Cac thang trong mot nam"<<'\n';
   for (thang=khongco; thang<=muoimot; thang++)
         cout<<"Day la thang: "<<thang<<'\n';
   getch();
   return 0;
 }
```

⇨ Trong chương trình này, do khai báo thành phần Khongco=1, nên trong vòng lặp for, chúng tôi cho biến thang chạy tới thành phần thứ muoimot để có kết quả đúng.

⇨ Chạy chương trình, bạn vẫn có kết quả như cũ.

▪ Bạn xem chương trình sau, và thử đoán xem kết quả trước khi bạn chạy thử chương trình.

**// Kieu_Liet_Ke.cpp**

```
#include<iostream.h>
#include<conio.h>
enum Traicay {cam,soai,oi,coc,chomchom,mangcut,saurieng};
int trai;
main()
 {
   clrscr();
   for (trai=cam; trai<=saurieng; trai++)
        switch (trai)
         {
           case 0:cout<<"Cam=0 \n";break;
           case 1:cout<<"Soai=1 \n";break;
           case 2:cout<<"Oi=2 \n";break;
           case 3:cout<<"Coc=3 \n";break;
           case 4:cout<<"Chom chom=4 \n";break;
           case 5:cout<<"Mang cut=5 \n";break;
           case 6:cout<<"Sau rieng=6 \n";break;
         }
  cout<<"soai-cam              = "<<soai-cam<<'\n';
  cout<<"oi-cam                = "<<oi-cam<<'\n';
  cout<<"saurieng-soai         = "<<saurieng-soai<<'\n';
  cout<<"cam+soai              = "<<cam+soai<<'\n';
  cout<<"cam-soai              = "<<cam-soai<<'\n';
  cout<<"cam-soai+saurieng     = "<<cam-soai+saurieng<<'\n';
  cout<<"saurieng-saurieng     = "<<saurieng-saurieng<<'\n';
  cout<<"cam+soai+oi+soai      = "<<cam+soai+oi+soai<<'\n';
  getch();
  return 0;
}
```

⇨ Trong chương trình trên, chúng tôi khai báo kiểu liệt kê có tên là Traicay, với các thành phần là cam, soai, oi, coc, chomchom, mangcut và saurieng. Vậy các giá trị số được gán sẽ là:

Cam=0
Soai=1
Oi=2
Coc=3
Chomchom=4
Mangcut=5
Saurieng=6

⇨ Chúng tôi khai báo biến trai có kiểu nguyên dùng làm biến chạy trong vòng lặp For.

⇨ Vòng lặp For và Switch sẽ cho ta biết các số tương ứng của các loại trái cây, nếu ta khai báo liệt kê như trên. Chúng tôi muốn chương trình sẽ in ra các giá trị này trước khi chúng tôi thực hiện các phép toán ở bên dưới.

⇨ Nếu chú ý, bạn thấy chúng tôi cho thành phần sau trừ thành phần trước để có số dương, thành phần trước trừ thành phần sau để có số âm, hai thành phần giống nhau trừ với nhau, cộng trừ hai thành phần giống nhau trong một biểu thức với mục đích để cho các bạn thấy các trường hợp để xem chương trình có báo như trong thực tế hay không.

# Bài 6 *(ngày 7)*

# Cấu trúc

▪ Các bạn thấy trong mảng và liệt kê ở trên các thành phần của chúng sẽ có cùng kiểu. Bây giờ chúng ta sẽ xem qua về cấu trúc trong C++

▪ Cấu trúc là tập hợp gồm những dữ liệu có kiểu khác nhau, được nhóm lại dưới một tên duy nhất cho tiện xử lý.

Ví dụ: Cấu trúc về phiếu liên lạc cho học sinh, mỗi "Phiếu liên lạc" được mô tả bởi tập các thuộc tính, chẳng hạn như họ tên, địa chỉ, lớp, điểm các môn, nhận xét ... Một số trong các thuộc tính này lại có thể là cấu trúc, đặc biệt chúng có thể là một hàm.

▪ Cấu trúc giúp cho việc tổ chức các dữ liệu phức tạp, đặc biệt trong những chương trình lớn, vì trong nhiều trường hợp, chúng cho phép nhóm các biến có liên quan lại để xử lý như một đơn vị thay vì như các thực thể tách biệt.

## ① Khai báo một cấu trúc

▪ Bắt đầu bằng từ khoá **struct**, đến tên kiểu của cấu trúc (tên này là một tên tuỳ chọn bất kỳ) sau đó là ngoặc nhọn mở, kế đến là các thành phần của cấu trúc, kế đến là dấu ngoặc nhọn đóng và dấu chấm phẩy.

*Ví dụ:*

```
struct giaovien
{
    char  Hoten[40];
    int   Namsinh;
    int   Namvaonganh;
    float Mucluong;
};
```

▪ Trong khai báo trên, bạn khai báo một cấu trúc có tên kiểu là giaovien gồm các thành phần là Hoten có kiểu là char, namsinh và Namvaonganh có kiểu int, Mucluong có kiểu float.

▪ Khi đã khai báo một cấu trúc, tên kiểu của cấu trúc sẽ là một kiểu, trong ví dụ trên, tên kiểu của cấu trúc là giaovien. Do đó giaovien sẽ là một kiểu.

▪ Bây giờ các bạn muốn khai báo một hoặc nhiều biến có kiểu là giaovien, các bạn cũng khai báo chúng như cách khai báo một biến bình thường trong C++.

*Ví dụ:*

▪ Muốn khai báo biến x có kiểu nguyên, chúng ta khai báo: Int x;

▪ Như vậy muốn khai báo biến GV1, GV2 có kiểu là giaovien, bạn khai báo như sau:

giaovien GV1;
giaovien GV2;

▪ Bạn cũng có thể khai báo các biến trực tiếp sau dấu ngoặc đóng của cấu trúc. Thay vì khai báo biến như trên, bạn có thể khai báo như sau :

```
struct giaovien
    {
        char  Hoten[40];
        int   Namsinh;
        int   Namvaonganh;
        float Mucluong;
    } GV1, GV2;
```

*Ghi chú:*

▪ Một khai báo cấu trúc không có danh sách các biến theo sau thì không được cấp phát bộ nhớ.Nó đơn thuần mô tả hình dáng của cấu trúc. Nếu khai báo có các biến theo sau thì sẽ được cấp phát không gian nhớ cho chúng.

### ② Truy xuất các thành phần của cấu trúc

▪ Để truy xuất các thành phần của cấu trúc, đầu tiên bạn khai báo biến có kiểu là tên kiểu của cấu trúc. Theo ví dụ trên bạn đã khai báo hai biến GV1 và GV2 có kiểu là giaovien. Bây giờ nếu bạn muốn truy xuất một thành phần nào của cấu trúc ứng với biến nào, bạn khai báo:

**Tên biến dấu chấm thành phần**

*Ví dụ:*

▪ Truy xuất tất cả các thành phần của biến GV1 ta viết:

GV1.Hoten;
GV1.Namsinh;

GV1.Namvaonganh;
GV1.Mucluong;

- Riêng biến GV2 chỉ cần truy xuất thành phần Hoten và Mucluong. Bạn viết như sau:

GV2.Hoten;
GV2.Mucluong;

*Ghi chú:* Trong chương trình, bạn khai báo cấu trúc trong hàm main(), khi các thành phần của cấu trúc có các kiểu bình thường. Khi có một thành phần của cấu trúc là một hàm, các bạn phải khai báo cấu trúc trước hàm main().

- Bạn có thể xem các ví dụ sau để hiểu rõ thêm về cấu trúc.

*Ví dụ 1:*

- Viết chương trình để nhập vào hai giáo viên, một giáo viên Toán và một giáo viên Văn, nội dung nhập gồm Họ và tên, năm sinh, năm vào ngành, mức lương, có gia đình hay chưa.

```
// cautruc_Nhap2Giaovien.cpp
#include<iostream.h>
#include<conio.h>
char hoi;
main()
 {
    struct giaovien {
          char Hoten[40];
          int namsinh;
          int namvaonganh;
          float mucluong;
          char cogiadinh;
    };
    giaovien GvToan;
    giaovien GvVan;
    cout<<"\n Nhap ho ten giao vien toan (khong nhap khoang trang):";
    cin>>GvToan.Hoten;
    cout<<"\n Nam sinh:";
    cin>>GvToan.namsinh;
    cout<<"\n Nam vao nganh:";
    cin>>GvToan.namvaonganh;
    cout<<"\n Muc luong:";
```

```
    cin>>GvToan.mucluong;
    cout<<"\n Co gia dinh c/k:";
    cin>>hoi;
    if (hoi=='c')
        GvToan.cogiadinh='c';
    else
        GvToan.cogiadinh='k';
    cout<<"\n Nhap ho ten giao vien van (khong nhap khoang trang:";
    cin>>GvVan.Hoten;
    cout<<"\n Nam sinh:";
    cin>>GvVan.namsinh;
    cout<<"\n Nam vao nganh:";
    cin>>GvVan.namvaonganh;
    cout<<"\n Muc luong:";
    cin>>GvVan.mucluong;
    cout<<"\n Co gia dinh c/k:";
    cin>>hoi;
    if (hoi=='c')
        GvVan.cogiadinh='c';
    else
        GvVan.cogiadinh='k';
    getch();
    return 0;
}
```

⇨ Trong chương trình trên, bạn khai báo cấu trúc có tên giaovien trong hàm main() với 5 thành phần là Hoten, Namsinh, Namvaonganh, Mucluong và Cogiadinh.

⇨ Sau đó bạn khai báo hai biến GvToan và GvVan có kiểu là giaovien.

⇨ Kế đến là bạn nhập vào các thành phần của GvToan và GvVan dùng tênbiến.thànhphần.

⇨ Ngoài cách khai báo biến có kiểu giaovien như trên, các bạn có thể khai báo như chúng tôi đã đề cập trong phần lý thuyết ở trên, có nghĩa là, thay vì dùng:

```
struct giaovien {
        char Hoten[40];
        int namsinh;
```

```
        int namvaonganh;
        float mucluong;
        char cogiadinh;
    };
    giaovien GvToan;
    giaovien GvVan;
```

- Ta có thể thay đoạn chương trình trên như sau nhưng vẫn có cùng ý nghĩa.

```
struct giaovien {
        char Hoten[40];
        int namsinh;
        int namvaonganh;
        float mucluong;
        char cogiadinh;
    } GvToan, GvVan ;
```

- Không có gì là khó phải không các bạn.

*Ghi chú:*

- Khi nhập họ tên của giáo viên, bạn không được có các khoảng trắng.

*Ví dụ 2:*

- Viết chương trình nhập vào hai giáo viên Toán và Văn như trên, sau đó cho xuất ra màn hình những thông tin của hai giáo viên vừa nhập.

```
// cautruc_Nhap_Xuat.cpp
#include<iostream.h>
#include<conio.h>
char hoi;
main()
 {
    struct giaovien {
        char Hoten[40];
        int namsinh;
        int namvaonganh;
        float mucluong;
        char cogiadinh;
    };
    giaovien GvToan;
    giaovien GvVan;
    cout<<"\n Nhap ho ten giao vien toan (khong nhap khoang trang): ";
```

```
cin>>GvToan.Hoten;
cout<<"\n Nam sinh: ";
cin>>GvToan.namsinh;
cout<<"\n Nam vao nganh: ";
cin>>GvToan.namvaonganh;
cout<<"\n Muc luong: ";
cin>>GvToan.mucluong;
cout<<"\n Co gia dinh c/k: ";
cin>>hoi;
if (hoi=='c')
    GvToan.cogiadinh='c';
else
    GvToan.cogiadinh='k';
cout<<"\n Nhap ho ten giao vien van (khong nhap khoang trang): ";
cin>>GvVan.Hoten;
cout<<"\n Nam sinh: ";
cin>>GvVan.namsinh;
cout<<"\n Nam vao nganh: ";
cin>>GvVan.namvaonganh;
cout<<"\n Muc luong: ";
cin>>GvVan.mucluong;
cout<<"\n Co gia dinh c/k: ";
cin>>hoi;
if (hoi=='c')
    GvVan.cogiadinh='c';
else
    GvVan.cogiadinh='k';
cout<<"\n          DANH SACH GIAO VIEN";
cout<<"\n ========================================";
cout<<"\n Ho ten giao vien toan: "<<GvToan.Hoten;
cout<<"\n Nam sinh giao vien toan: "<<GvToan.namsinh;
cout<<"\n Nam vao nganh giao vien toan: "<<GvToan.namvaonganh;
cout<<"\n Muc luong giao vien toan: "<<GvToan.mucluong;
cout<<"\n Co gia dinh: "<<GvToan.cogiadinh;
cout<<"\n Ho ten giao vien van: "<<GvVan.Hoten;
cout<<"\n Nam sinh giao vien van: "<<GvVan.namsinh;
cout<<"\n Nam vao nganh giao vien van: "<<GvVan.namvaonganh;
cout<<"\n Muc luong giao vien van: "<<GvVan.mucluong;
```

```
        cout<<"\n Co gia dinh: "<<GvVan.cogiadinh;
        cout<<"\n ========================================";
        getch();
        return 0;
}
```

⇨ Phần đầu của chương trình này cũng giống như chương trình trên, Phần sau là xuất các thông tin ra màn hình, việc xuất này cũng đơn giản, không có gì khó cả, các bạn tự tìm hiểu.

⇨ Chạy thử chương trình trên, bạn nhập họ tên giáo viên toán (lưu ý không có khoảng trắng), gõ Enter để nhập tiếp Năm sinh ... Sau khi nhập xong các nội dung của hai giáo viên, chương trình sẽ in ra trên màn hình những thông tin mà bạn vừa nhập. Bạn chạy thử để thấy kết quả.

*Ví dụ 3:*

▪ Giống như ví dụ 2 ở trên, nhưng trong cấu trúc giaovien có thêm một hàm noiquy dùng để thông báo nội quy cho giáo viên. Như đã nói ở trên, vì một thành phần có dùng hàm, do đó chúng ta phải khai báo cấu trúc ở bên ngoài hàm main(). Bạn xem ví dụ sau:

**// cautruc_coHam.cpp**

```
#include<iostream.h>
#include<conio.h>
char hoi;
    struct giaovien {
            char Hoten[40];
            int namsinh;
            int namvaonganh;
            float mucluong;
            char cogiadinh;
            public:void noiquy (void);
    };
    void giaovien::noiquy(void)
        {
            cout<<"\n GIAO VIEN PHAI CHAP HANH TOT NOI QUY SAU";
            cout<<"\n Khong duoc nghi khong co ly do";
            cout<<"\n Phai xin phep truoc va goi lai giao an";
        }
main()
 {
```

```
giaovien quydinh;
giaovien GvToan;
giaovien GvVan;
cout<<"\n Nhap ho ten giao vien toan (khong nhap khoang trang): ";
cin>>GvToan.Hoten;
cout<<"\n Nam sinh: ";
cin>>GvToan.namsinh;
cout<<"\n Nam vao nganh: ";
cin>>GvToan.namvaonganh;
cout<<"\n Muc luong: ";
cin>>GvToan.mucluong;
cout<<"\n Co gia dinh c/k: ";
cin>>hoi;
if (hoi=='c')
    GvToan.cogiadinh='c';
else
    GvToan.cogiadinh ='k';
cout<<"\n Nhap ho ten giao vien van (khong nhap khoang trang): ";
cin>>GvVan.Hoten;
cout<<"\n Nam sinh: ";
cin>>GvVan.namsinh;
cout<<"\n Nam vao nganh: ";
cin>>GvVan.namvaonganh;
cout<<"\n Muc luong: ";
cin>>GvVan.mucluong;
cout<<"\n Co gia dinh c/k: ";
cin>>hoi;
if (hoi=='c')
    GvVan.cogiadinh='c';
else
    GvVan.cogiadinh='k';
cout<<"\n        DANH SACH GIAO VIEN";
cout<<"\n ========================================";
cout<<"\n Ho ten giao vien toan: "<<GvToan.Hoten;
cout<<"\n Nam sinh giao vien toan: "<<GvToan.namsinh;
cout<<"\n Nam vao nganh giao vien toan : "<<GvToan.namvaonganh;
cout<<"\n Muc luong giao vien toan: "<<GvToan.mucluong;
cout<<"\n Co gia dinh: "<<GvToan.cogiadinh;
```

```
    cout<<"\n Ho ten giao vien van: "<<GvVan.Hoten;
    cout<<"\n Nam sinh giao vien van: "<<GvVan.namsinh;
    cout<<"\n Nam vao nganh giao vien van: "<<GvVan.namvaonganh;
    cout<<"\n Muc luong giao vien van: "<<GvVan.mucluong;
    cout<<"\n Co gia dinh: "<<GvVan.cogiadinh;
    quydinh.noiquy();
    cout<<"\n ==========================================";
    getch();
    return 0;
}
```

# Bài 7 *(ngày 8)*

# Hợp – Hợp không tên

▪ Hợp cũng giống như cấu trúc, nhưng các thành phần của hợp dùng chung một vùng bộ nhớ riêng biệt, không bao hàm bất kỳ thông tin phụ thuộc máy nào trong chương trình. Kích thước của hợp là kích thước của thành phần lớn nhất ở trong hợp.

## ① Khai báo

▪ Bắt đầu bằng từ khoá union, đến tên kiểu của hợp (tên này là một tên tuỳ chọn bất kỳ) sau đó là ngoặc nhọn mở, kế đến là các thành phần của hợp, kế đến là dấu ngoặc nhọn đóng và dấu chấm phẩy.

*Ví dụ:*

```
Union   kieuso
    {
        int       songuyen;
        float     sothuc;
        double sothucdai;
    };
```

▪ Trong khai báo trên, bạn khai báo một hợp có tên là kieuso gồm các thành phần là songuyen có kiểu là int, sothuc co kiểu là float và sothucdai có kiểu double.

▪ Khi đã khai báo một hợp, tên kiểu của hợp sẽ là một kiểu, trong ví dụ trên, tên kiểu của hợp là kieuso. Do đó kieuso sẽ là một kiểu.

▪ Bây giờ các bạn muốn khai báo một hoặc nhiều biến có kiểu là kieuso, các bạn cũng khai báo chúng như cách khai báo một biến bình thường

*Ví dụ:*

▪ Muốn khai báo biến x có kiểu nguyên, chúng ta khai báo: Int x;

▪ Muốn khai báo biến KS1, KS2 có kiểu là Kieuso, bạn khai báo như sau:

kieuso KS1;

kieuso KS2;

• Bạn cũng có thể khai báo các biến trực tiếp sau dấu ngoặc đóng của hợp. Thay vì khai báo biến như trên, bạn có thể khai báo như sau:

```
Union kieuso
    {
        int       songuyen;
        float     sothuc;
        double sothucdai;
    } KS1, KS2;
```

## ② Truy xuất các thành phần của hợp

• Để truy xuất các thành phần của hợp, đầu tiên bạn khai báo biến có kiểu là tên kiểu của hợp. Theo ví dụ trên bạn đã khai báo hai biến KS1 và KS2 có kiểu là kieuso. Bây giờ nếu bạn muốn truy xuất một thành phần nào của hợp ứng với biến nào, bạn khai báo:

**Tên biến  dấu chấm  thành phần**

*Ví dụ:*

• Truy xuất tất cả các thành phần của biến KS1 ta viết:

```
KS1.songuyen;
KS1.sothuc;
KS1.sothucdai;
```

• Riêng biến KS2 chỉ cần truy xuất thành phần songuyen và sothuc. Bạn viết như sau:

```
KS2.songuyen;
KS2.sothuc;
```

*Ghi chú:*

• Trong chương trình, bạn khai báo hợp trong hàm main(), khi các thành phần của hợp có các kiểu bình thường. Khi có một thành phần của hợp là một hàm, các bạn phải khai báo hợp trước hàm main().

• Bạn có thể xem một vài ví dụ sau để hiểu rõ hơn về hợp

*Ví dụ 1:*

• Chúng tôi viết chương trình này để cho các bạn thấy rằng, kích thước của hợp sẽ là kích thước của thành phần lớn nhất chứa trong hợp, còn trong cấu trúc, kích thước của nó sẽ bằng tổng kích thước của các thành phần.

**// Kichthuoc_Hop.cpp**

```
#include<iostream.h>
#include<conio.h>
main()
 {
        union Kichthuochop {
           int songuyen;
           float sothuc;
           double sothucdai;
        };
    clrscr();
    cout<<"\n Kich thuoc cua union tren la: " << "Byte";
    cout<<sizeof(union Kichthuochop);
    getch();
    return 0;
 }
```

- Trong union trên, bạn khai báo 3 thành phần là songuyen có kiểu int, sothuc có kiểu là float, sothucdai có kiểu là double.
- Chạy thử chương trình trên, bạn thấy kết quả thông báo như sau:

Kích thước của union trên là: 8 byte.

- Như bạn đã biết kiểu int có kích thước là 2 byte, kiểu float có kích thước là 4 byte và kiểu double có kích thước là 8 byte. Do đó chương trình sẽ báo ra kích thước của thành phần lớn nhất.
- Nếu bây giờ bạn xoá thành phần sothucdai trong union của chương trình, bạn sẽ có kết quả báo ra là:

Kích thước của union trên là: 4 byte.

- Bạn chạy chương trình sau để biết kích thước của cấu trúc

*Ví dụ 2:*

**// Kichthuoc_cautruc.cpp**

```
#include<iostream.h>
#include<conio.h>
main()
 {
    struct Kichthuoccautruc {
           int songuyen;
          float sothuc;
          double sothucdai;
    };
```

```
    clrscr();
    cout<<"\n Kich thuoc cua union tren la: ";
    cout<<sizeof(struct Kichthuoccautruc)<<"Byte";
    getch();
    return 0;
}
```

- Chạy chương trình, bạn sẽ thấy báo kết quả: Kích thước của struct trên là: 14 byte.
- Nếu xoá thành phần sothucdai trong cấu trúc, chạy chương trình, bạn sẽ có kết quả: Kích thước của struct trên là: 6 byte.
- Qua hai chương trình trên, chúng tôi cho các bạn thấy về kích thước sử dụng để các bạn áp dụng struct hay union cho các trường hợp cho phù hợp.

*Ví dụ 3:*

- Viết chương trình nhập vào hai giáo viên, một giáo viên Toán và một giáo viên Văn, sau đó in thông tin vừa nhập của hai giáo viên ra màn hình.

**// Cautruc_Hop.cpp**

```
#include<iostream.h>
#include<conio.h>
char hoi;
main()
{
   union giaovien {
         char Hoten[40];
         int namsinh;
         int namvaonganh;
         float mucluong;
         char cogiadinh;
   };
   giaovien GvToan;
   giaovien GvVan;
   cout<<"\n Nhap ho ten giao vien toan (khong nhap khoang trang): ";
   cin>>GvToan.Hoten;
   cout<<"\n Nam sinh: ";
   cin>>GvToan.namsinh;
   cout<<"\n Nam vao nganh: ";
   cin>>GvToan.namvaonganh;
```

```
    cout<<"\n Muc luong: ";
    cin>>GvToan.mucluong;
    cout<<"\n Co gia dinh c/k: ";
    cin>>hoi;
    if (hoi=='c')
        GvToan.cogiadinh='c';
    else
        GvToan.cogiadinh='k';
    cout<<"\n Nhap ho ten giao vien van (khong nhap khoang trang): ";
    cin>>GvVan.Hoten;
    cout<<"\n Nam sinh: ";
    cin>>GvVan.namsinh;
    cout<<"\n Nam vao nganh: ";
    cin>>GvVan.namvaonganh;
    cout<<"\n Muc luong: ";
    cin>>GvVan.mucluong;
    cout<<"\n Co gia dinh c/k: ";
    cin>>hoi;
    if (hoi=='c')
        GvVan.cogiadinh='c';
    else
        GvVan.cogiadinh='k';
    cout<<"\n          DANH SACH GIAO VIEN";
    cout<<"\n ========================================";
    cout<<"\n Ho ten giao vien toan: "<<GvToan.Hoten;
    cout<<"\n Nam sinh giao vien toan: "<<GvToan.namsinh;
    cout<<"\n Nam vao nganh giao vien toan: "<<GvToan.namvaonganh;
    cout<<"\n Muc luong giao vien toan: "<<GvToan.mucluong;
    cout<<"\n Co gia dinh: "<<GvToan.cogiadinh;
    cout<<"\n Ho ten giao vien van: "<<GvVan.Hoten;
    cout<<"\n Nam sinh giao vien van: "<<GvVan.namsinh;
    cout<<"\n Nam vao nganh giao vien van: "<<GvVan.namvaonganh;
    cout<<"\n Muc luong giao vien van: "<<GvVan.mucluong;
    cout<<"\n Co gia dinh: "<<GvVan.cogiadinh;
    cout<<"\n ========================================";
    getch();
    return 0;
}
```

- Nó chỉ khác ở cấu trúc là từ khoá union thay cho từ khoá struct.
- Thay vì khai báo biến sau, bạn có thể khai báo trực tiếp sau dấu ngoặc nhọn đóng, ví dụ:

```
union giaovien {
        char Hoten[40];
        int namsinh;
        int namvaonganh;
        float mucluong;
        char cogiadinh;
    };
    giaovien GvToan;
    giaovien GvVan;
```

- Chúng ta có thể thay lại đoạn chương trình trên như sau :

```
union giaovien {
        char Hoten[40];
        int namsinh;
        int namvaonganh;
        float mucluong;
        char cogiadinh;
    } GvToan,GvVan;
```

*Ví dụ 4:*

- Viết chương trình dùng union giống như trên nhưng có thêm hàm để thông báo nội quy. Vì có hàm nên bạn phải khai báo union ở bên ngoài hàm main();

```
// cautruc_Hop.cpp
#include<iostream.h>
#include<conio.h>
char hoi;
    union giaovien {
        char Hoten[40];
        int namsinh;
        int namvaonganh;
        float mucluong;
        char cogiadinh;
        public:void noiquy (void);
    };
    void giaovien::noiquy(void)
```

```
        {
            cout<<"\n GIAO VIEN PHAI CHAP HANH TOT NOI QUY SAU";
            cout<<"\n Khong duoc nghi khong co ly do";
            cout<<"\n Phai xin phep truoc va goi lai giao an";
        }
main()
 {
    giaovien quydinh;
    giaovien GvToan;
    giaovien GvVan;
    cout<<"\n Nhap ho ten giao vien toan (khong nhap khoang trang): ";
    cin>>GvToan.Hoten;
    cout<<"\n Nam sinh: ";
    cin>>GvToan.namsinh;
    cout<<"\n Nam vao nganh: ";
    cin>>GvToan.namvaonganh;
    cout<<"\n Muc luong: ";
    cin>>GvToan.mucluong;
    cout<<"\n Co gia dinh c/k: ";
    cin>>hoi;
    if (hoi=='c')
        GvToan.cogiadinh='c';
    else
        GvToan.cogiadinh='k';
    cout<<"\n Nhap ho ten giao vien van (khong nhap khoang trang): ";
    cin>>GvVan.Hoten;
    cout<<"\n Nam sinh: ";
    cin>>GvVan.namsinh;
    cout<<"\n Nam vao nganh: ";
    cin>>GvVan.namvaonganh;
    cout<<"\n Muc luong: ";
    cin>>GvVan.mucluong;
    cout<<"\n Co gia dinh c/k: ";
    cin>>hoi;
    if (hoi=='c')
        GvVan.cogiadinh='c';
    else
        GvVan.cogiadinh='k';
```

```
    cout<<"\n          DANH SACH GIAO VIEN";
    cout<<"\n ========================================";
    cout<<"\n Ho ten giao vien toan: "<<GvToan.Hoten;
    cout<<"\n Nam sinh giao vien toan: "<<GvToan.namsinh;
    cout<<"\n Nam vao nganh giao vien toan: "<<GvToan.namvaonganh;
    cout<<"\n Muc luong giao vien toan: "<<GvToan.mucluong;
    cout<<"\n Co gia dinh: "<<GvToan.cogiadinh;
    cout<<"\n Ho ten giao vien van: "<<GvVan.Hoten;
    cout<<"\n Nam sinh giao vien van: "<<GvVan.namsinh;
    cout<<"\n Nam vao nganh giao vien van: "<<GvVan.namvaonganh;
    cout<<"\n Muc luong giao vien van: "<<GvVan.mucluong;
    cout<<"\n Co gia dinh: "<<GvVan.cogiadinh;
    quydinh.noiquy();
    cout<<"\n ========================================";
    getch();
    return 0;
}
```

- Chạy thử chương trình này, bạn sẽ thấy kết quả giống như trong chương trình ở phần cấu trúc.
- Viết chương trình dùng hợp nhưng không có tên kiểu hợp, không có tên biến kiểu hợp.

*Ví dụ 5:*

```
// cautruc_Hop_khongten.cpp
#include<iostream.h>
#include<conio.h>
char hoi;
main()
 {
   union {
         char Hoten[40];
         int namsinh;
         int namvaonganh;
         float mucluong;
         char cogiadinh;
   };
   cout<<"\n Nhap ho ten giao vien toan (khong nhap khoang trang): ";
   cin>>Hoten;
   cout<<"\n Nam sinh: ";
```

```
    cin>>namsinh;
    cout<<"\n Nam vao nganh: ";
    cin>>namvaonganh;
    cout<<"\n Muc luong: ";
    cin>>mucluong;
    cout<<"\n Co gia dinh c/k: ";
    cin>>hoi;
    if (hoi=='c')
        cogiadinh='c';
    else
        cogiadinh='k';

    cout<<"\n          DANH SACH GIAO VIEN";
    cout<<"\n ========================================";
    cout<<"\n Ho ten giao vien: "<<Hoten;
    cout<<"\n Nam sinh giao vien: "<<namsinh;
    cout<<"\n Nam vao nganh giao vien: "<<namvaonganh;
    cout<<"\n Muc luong giao vien: "<<mucluong;
    cout<<"\n Co gia dinh: "<<cogiadinh;
    cout<<"\n ========================================";
    getch();
    return 0;
}
```

▪ Khi không dùng tên kiểu hợp và tên biến kiểu hợp, bạn sẽ truy xuất bằng tên các thành phần như trên.

# Bài 8 *(ngày 9)*

# Biến trỏ

• Trong chương trình, khi các bạn khai báo một biến, biến đó sẽ có một địa chỉ duy nhất trong bộ nhớ, chỉ rõ nơi bắt đầu của phần bộ nhớ chiếm bởi giá trị của biến. Không gian kí ức dành cho biến tuỳ thuộc vào kiểu của biến, các bạn có thể thấy kích thước của một số biến, của union và struct trong ví dụ 1 và ví dụ 2 của phần hợp vừa nêu ở trên.

• Vì dữ liệu được chứa trong bộ nhớ theo một thứ tự nhất định, do đó chúng ta có thể tiếp cận các dữ liệu này, bằng cách thao tác một biến chứa địa chỉ thích hợp đó là dùng biến con trỏ.

• Con trỏ là biến chứa địa chỉ của một biến khác. Vì con trỏ chứa địa chỉ của một đối tượng nào đó, nên có thể xâm nhập vào đối tượng thông qua con trỏ.

## ① Khai báo

• Khi cần dùng con trỏ, các bạn sẽ khai báo như sau:

**Kiểu *Têncontro**

*Ví dụ:*

```
int  *xcontro;
float  *ycontro;
char  *Gvcontro;
```

• Chúng ta có thể khai báo một biến con trỏ cho mọi đối tượng trong bộ nhớ, như chuỗi, mảng, cấu trúc, hàm và ngay cả một biến con trỏ khác.

## ② Truy xuất

• Để truy xuất giá trị của một biến con trỏ ta dùng dấu * trước tên biến con trỏ, ví dụ như *Gvcontro, *xcontro, *ycontro.

• Để biết địa chỉ của một biến, ta dùng phép toán một ngôi & trước tên biến. Ví dụ như &x, &y, &Gvcontro, &xcontro, &ycontro...

• Bạn có thể xem các ví dụ sau để hiểu rõ hơn về con trỏ.

*Ví dụ 1:*

Viết chương trình thông báo kích thước của các biến, giá trị, địa chỉ ... của các biến.

**// Contro.cpp**

```
#include<iostream.h>
#include<conio.h>
main()
 {
    clrscr();
    int x;
    double y;
    float z;
    char hoi;
    x=200;
    y=12.25;
    z=2.45;
    int *xcontro;
    double *ycontro;
    float *zcontro;
    cout<<"\n Muon biet dia chi cua x ban dung &x. &x= "<<&x;
    cout<<"\n kich thuoc chua bien x la: "<<sizeof(x)<<"Byte";
    cout<<"\n Muon biet dia chi cua y ban dung &y. &y= "<<&y;
    cout<<"\n kich thuoc chua bien y la: "<<sizeof(y)<<" Byte";
    cout<<"\n Muon biet dia chi cua z ban dung &z. &z= "<<&z;
    cout<<"\n kich thuoc chua bien z la: "<<sizeof(z)<<" Byte";
    cout<<"\n Muon biet dia chi cua xcontro ban dung &xcontro. &xcontro =
            "<<&xcontro;
    cout<<"\n kich thuoc chua bien xcontro la: "<<sizeof(xcontro)<<" Byte";
    cout<<"\n Muon biet dia chi cua ycontro ban dung &ycontro. &ycontro =
           "<<&ycontro;
    cout<<"\n kich thuoc chua bien ycontro la: "<<sizeof(ycontro)<<" Byte";
    cout<<"\n Muon biet dia chi cua zcontro ban dung &zcontro. &zcontro =
           "<<&zcontro;
    cout<<"\n kich thuoc chua bien zcontro la: "<<sizeof(zcontro)<<" Byte";
    cout<<"\n Noi dung cua bien x la: "<<x;
    cout<<"\n Noi dung cua bien y la: "<<y;
    cout<<"\n Noi dung cua bien z la: "<<z;
    xcontro=&x;
    ycontro=&y;
```

```
    zcontro=&z;
    cout<<"\n Dia chi ma xcontro chi den la: "<<xcontro;
    cout<<"\n Gia tri cua xcontro chi den la: "<<*xcontro;
    cout<<"\n Dia chi ma ycontro chi den la: "<<ycontro;
    cout<<"\n Gia tri cua ycontro chi den la: "<<*ycontro;
    cout<<"\n Dia chi ma zcontro chi den la: "<<zcontro;
    cout<<"\n Gia tri cua zcontro chi den la: "<<*zcontro;
    getch();
    return 0;
}
```

▪ Chạy thử chương trình trên, nhìn vào các kết quả báo ra, bạn sẽ hiểu rõ những điều chúng tôi đã đề cập ở trên. Các bạn có thể thay đổi một chút để học hỏi thêm.

### ③ Biến con trỏ áp dụng trong chuỗi

▪ Ta biết trong C++, chuỗi cũng giống như một mảng, để truy xuất các kí tự trong một chuỗi ta dùng chỉ số của chúng. Ví dụ bạn khai báo một chuỗi như sau:

Char dongho[50];

Và bạn nhập vào Treotuong

Vậy giá trị của dongho[0] là kí tự 'T'

Dongho[1] là kí tự 'r'

.....

.....

Dongho[8] là kí tự 'g'

▪ Tuy nhiên, bạn có thể dùng biến con trỏ để trỏ đến các kí tự của một chuỗi để truy cập đến các phần tử trong chuỗi. Bạn có thể xem ví dụ sau:

*Ví dụ 2:*

**// contro_va_chuoi.cpp**

```
#include<iostream.h>
#include<conio.h>
#include<string.h>

main()
  {
      char chuoi[100];
      char *chuoicontro;
      chuoicontro = chuoi;
```

```
        int i;
        clrscr();
        cout<<"\n Ban nhap vao chuoi: ";
        cin>>chuoi;
        for (i=0; i<=strlen(chuoi)-1; i++)
            cout<<"\n Ki tu thu "<<i<<"la: "<<*chuoicontro++;
        getch();
        return 0;
    }
```

- Chương trình trên sẽ cho bạn nhập vào một chuỗi, sau đó in ra các kí tự của chuỗi, đồng thời cũng cho biết vị trí của các kí tự.

*Ví dụ 3:*

**// contro_va_chuoi.cpp**

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
main()
  {
      char chuoi[100];
      char *chuoicontro;
      chuoicontro=chuoi;
      int i;
      int sokitu;
      clrscr();
      cout<<"\n Ban nhap vao chuoi: ";
      cin>>chuoi;
      cout<<"\n Ban can xem bao nhieu ki tu? so ki tu= ";
      cin>>sokitu;
      for (i=0; i<sokitu; i++)
          cout<<"n Ki tu thu "<<i<<" la: "<<*chuoicontro++;

      getch();
      return 0;
  }
```

- Chương trình này sẽ cho in ra số kí tự trong chuỗi theo yêu cầu của người sử dụng.

### ④ Biến con trỏ dùng trong cấu trúc

▪ Qua chương về cấu trúc, các bạn đã biết cách truy xuất một thành phần nằm trong cấu trúc là

**Tênbiếncókiểucấutrúc.Tênthànhphần**

▪ Nếu bạn khai báo cấu trúc giaovien như sau:

```
Struct  giaovien
        {
          char  Hoten[40];
          int     Namsinh;
          float  Mucluong;
        };
giaovien  *GvToan;
giaovien  *GvVan;
```

▪ Muốn truy xuất thành phần Hoten của biến GvToan, bạn dùng:
GvToan.Hoten;

▪ Muốn truy xuất thành phần Hoten của biến con trỏ GvVan, bạn phải dùng:
GvVan -> Hoten;

*Ghi chú*:

▪ Chỉ khác là trong biến con trỏ không dùng dấu chấm, mà phải dùng dấu mũi tên (bạn đánh dấu – và dấu >).

▪ Bạn có thể xem ví dụ sau để thấy những điều vừa nói ở trên:

Ví dụ 4:

```
// Contro_cautruc.cpp
 #include<iostream.h>
 #include<conio.h>
 #include<string.h>

struct giaovien
        {
          char Hoten[40];
          char Namsinh[4];
          char Namvaonganh[4];
          char Mucluong[10];
        };
giaovien *GVToan;
giaovien *GVVan;
```

```
main()
 {
    clrscr();
    char TenToan[40], TenVan[40];
    char NSToan[4], NSVan[4];
    char NVNToan[4], NVNVan[4];
    char MLToan[10], MLVan[10];

    cout<<"\n Ban nhap Ho ten giao vien Toan: ";
    cin>>GVToan -> Hoten;
    strcpy(TenToan,GVToan -> Hoten);
    cout<<"\n Nam sinh cua giao vien Toan: ";
    cin>>GVToan -> Namsinh;
    strcpy(NSToan,GVToan -> Namsinh);
    cout<<"\n Nam vao nganh cua giao vien Toan: ";
    cin>>GVToan -> Namvaonganh;
    strcpy(NVNToan,GVToan -> Namvaonganh);
    cout<<"\n Muc luong giao vien Toan: ";
    cin>>GVToan -> Mucluong;
    strcpy(MLToan,GVToan -> Mucluong);

    cout<<"\n Ban nhap Ho ten giao vien Van: ";
    cin>>GVVan -> Hoten;
    strcpy(TenVan,GVVan -> Hoten);
    cout<<"\n Nam sinh cua giao vien Van: ";
    cin>>GVVan -> Namsinh;
    strcpy(NSVan,GVVan -> Namsinh);
    cout<<"\n Nam vao nganh cua giao vien Van: ";
    cin>>GVVan -> Namvaonganh;
    strcpy(NVNVan,GVVan -> Namvaonganh);
    cout<<"\n Muc luong giao vien Van: ";
    cin>>GVVan -> Mucluong;
    strcpy(MLVan,GVVan -> Mucluong);

    cout<<"\n Ho va ten giao vien Toan "<<TenToan;
    cout<<"\n Nam sinh: "<<NSToan;
    cout<<"\n Nam vao nganh: "<<NVNToan;
    cout<<"\n Muc luong: "<<MLToan;
```

```
    cout<<"\n Ho va ten giao vien Van "<<TenVan;
    cout<<"\n Nam sinh: "<<NSVan;
    cout<<"\n Nam vao nganh: "<<NVNVan;
    cout<<"\n Muc luong: "<<MLVan;

  getch();
  return 0;
}
```

- Chạy thử chương trình để thấy kết quả.

# Bài 9 *(ngày 10)*

# Tập tin

▪ Qua các chương trình ở trên ta thấy khi nhập vào thông tin và sau đó xuất chúng ra màn hình, sau khi thực hiện xong, các thông tin sẽ mất luôn, không tồn tại để cho các bạn có thể truy xuất chúng ở những lần sau. Trong bài này, các bạn sẽ thấy cách dùng kiểu tập tin dạng văn bản hoặc dạng nhị phân, kết hợp với biến con trỏ để thực hiện việc đọc và ghi các tập tin,

① **Các bước để tạo và đọc tập tin trên đĩa**

- Dùng tập tin tiêu đề #include<stdio.h>
- Khai báo biến trỏ kiểu FILE (Bạn phải dùng kí tự in hoa)
- Khai báo đối tượng nhận dữ liệu
- Mở tập tin
- Dùng các hàm thích hợp để đọc hoặc viết dữ liệu
- Đóng tập tin lại

② **Khai báo biến trỏ kiểu FILE**

**FILE *Tencontro**

▪ Bạn có thể xem một số ví dụ sau:

☞ **File văn bản**

*Ví dụ 1:*

▪ Viết chương trình để viết vào một file văn bản.

```
// Nhap_Taptinvanban.cpp
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
main()
 {
   clrscr();
```

```
    FILE *taptin;
    char c;
    taptin = fopen("c:\\vanban.Txt","wt");
    while ((c=getchar())!=EOF)
     {
        fputc(c,taptin);
     }
    fclose(taptin);
    return 0;
};
```

⇨ Trong chương trình này, các bạn dùng các hàm fopen, fputc, fclose, và getchar.

⇨ Hàm fopen với hai đối số, đối số đầu tiên sẽ là đường dẫn và tên của tập tin văn bản mà bạn cần ghi. Trong chương trình trên là "C:\\vanban.Txt". Đối số thứ hai là kí tự nhận biết để thực hiện. Trong chương trình trên là "wt", ý nghĩa của nó là viết một tập tin văn bản (bạn có thể nhớ w là write, t là text).

⇨ Hàm fputc sẽ xuất một kí tự vào luồng.

⇨ Hàm fclose dùng để đóng tập tin.

⇨ Chương trình trên sẽ nhận các kí tự do người sử dụng nhập từ bàn phím, không muốn nhập nữa thì nhấn phím F6.

*Ví dụ 2:*

• Viết chương trình để xuất ra màn hình tập tin văn bản mà bạn vừa nhập ở trên.

```
// Xuat_Taptinvanban.cpp
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
main()
 {
    clrscr();
    FILE *taptin;
    char c;
    taptin = fopen("c:\\vanban.Txt","rt");
    while (!feof(taptin))
     {
```

```
        c=fgetc(taptin);
        putchar(c);
      }
    fclose(taptin);
    getch();
    return 0;
  };
```

- Chương trình trên sẽ đọc tập tin có tên vanban.Txt nằm trong thư mục gốc của ổ đĩa C. Đối số thứ hai của hàm fopen là "rt" (bạn có thể nhớ r là read, và t là text ).
- Nếu bạn có các tập tin dạng văn bản khác nằm ở trên đĩa, bạn có thể khai báo đường dẫn và tên tập tin ở vị trí đối số một của hàm fopen.

### ☞ File nhị phân

*Ví dụ 1:*

- Chương trình viết một File nhị phân

**// Taptin_Nhiphan_Viet.cpp**

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
 {
    clrscr();
    struct giaovien
            {
              char Hoten[40];
              int  Namsinh;
              int  Namvaonganh;
              int Mucluong;
            };
    giaovien GV;
    FILE *taptin;
    if ((taptin = fopen("c:\\GIAOVIEN.DAT","wb"))==NULL)
        {
          cout<<"Khong the mo tap tin GIAOVIEN.DAT";
          exit(1);
        }
```

```
    do
     {
        cout<<"\n Ho va ten cua giao vien: ";
        cin>>GV.Hoten;
        cout<<"\n Nam sinh: ";
        cin>>GV.Namsinh;
        cout<<"\n Nam vao nganh: ";
        cin>>GV.Namvaonganh;
        cout<<"\n Muc luong: ";
        cin>>GV.Mucluong;
        fwrite(&GV,sizeof(GV),1,taptin);
        cout<<"\n Co nhap them giao vien khong (c/k)?";
     }
     while (getche()=='c');
    fclose(taptin);
    return 0;
 }
```

⇨ Chương trình này sẽ viết vào tập tin có tên GIAOVIEN.DAT ở thư mục gốc của ổ đĩa C. Đối số thứ hai của hàm fopen là "wb" (bạn có thể nhớ w là write, b là binary).

⇨ Chương trình này dùng cấu trúc và con trỏ kiểu FILE.

⇨ Chương trình sau để đọc tập tin nhị phân vừa nhập ở trên.

*Ví dụ 2:*

**// Taptin_Nhiphan_Doc.cpp**

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
 {
    clrscr();
    struct giaovien
            {
              char Hoten[40];
              int  Namsinh;
              int  Namvaonganh;
              int Mucluong;
```

```
            };
    giaovien GV;
    FILE *taptin;
    if ((taptin = fopen("c:\\GIAOVIEN.DAT","rb"))==NULL)
        {
          cout<<"Khong the mo tap tin GIAOVIEN.DAT";
          exit(1);
        }
    while (fread(&GV,sizeof(GV),1,taptin)==1)
      {
          cout<<"\n Ho va ten cua giao vien : "<<GV.Hoten;
          cout<<"\n Nam sinh : "<<GV.Namsinh;
          cout<<"\n Nam vao nganh : "<<GV.Namvaonganh;
          cout<<"\n Muc luong : "<<GV.Mucluong;
      }
    fclose(taptin);
    getch();
    return 0;
}
```

▪ Bạn lưu ý đối số thứ hai của hàm fopen trong trường hợp đọc là "rb" (r là read, b là binary).

# Bài 10 *(ngày 11, 12 và 13)*

# Đóng gói, thừa kế Đa hình trong C++

▪ Trong các chương trước, các bạn đã thực tập và viết các chương trình trên C++, nó chỉ hơi khác một chút về các tập tin tiêu đề, các câu lệnh nhập và xuất, một số hàm so với ngôn ngữ C... và một số vấn đề khác. Do đó nó cũng không khó với những bạn đã biết về C.

▪ Trong chương này, chúng tôi giới thiệu thêm về một số khái niệm trong **lập trình hướng đối tượng**, đây là nét đặc thù của C++.

▪ Biết nói thế nào để bạn có thể hình dung và có thể hiểu ngay về lập trình hướng đối tượng, chúng tôi xin phác họa để các bạn có thể thấy ngay về nó, đó là các bạn hãy xem trò chơi lắp ráp của con cháu mình. Đến siêu thị hoặc các cửa hàng bán trò chơi cho các cháu, chắc các bạn thấy bán rất nhiều, có khi còn cho các cháu vào lắp ráp, cháu nào tạo được hình đẹp thì sẽ được phần thưởng. Đồ chơi gồm những miếng nhựa với các hình thù khác nhau, trên các miếng sẽ có lỗ hoặc những thanh nhỏ để ghép nối lại, các bạn có thể hình dung ra các dụng cụ này là những đối tượng đã có sẵn, từ những đối tượng này, mỗi người chúng ta có thể tạo ra những hình khác nhau, người thì tạo ra một chiếc xe, người thì tạo ra một ngôi nhà, người thì tạo ra một người máy robo ... Như vậy các bạn có thể hình dung ra từ một đối tượng có hình dạng cố định như vậy, dưới các hình, nó có thể đóng các vai trò khác nhau, con người có thể sử dụng nó cho công việc của mình nếu như thích hợp. Ở đây chỉ nói lên cấu trúc bên ngoài, khi lập trình bạn cần hiểu rõ đối tượng đó về công dụng và khả năng tác động của nó, hiểu càng rõ thì các bạn càng sử dụng chúng một cách có hiệu quả. Theo chúng tôi ví dụ này chắc sẽ giúp các bạn hình dung ra về lập trình hướng đối tượng.

▪ Trong lập trình hướng đối tượng, các bạn cần nắm những khái niệm mới như:

- **Sự đóng gói (Encapsulation)**
- **Sự thừa kế (Inheritance)**
- **Sự đa hình (Polymorphism)**

▪ Bạn có thể hình dung các khái niệm này như sau:

✍ Sự đóng gói (Encapsulation) là sự liên kết các hàm (Phương thức) trong đó có liên kết cấu trúc dữ liệu để thao tác dữ liệu. Sự đóng gói này trong C++ cũng như một số ngôn ngữ lập trình hướng đối tượng khác được thực hiện bằng một kiểu dữ liệu mới. Đó là kiểu lớp.

✍ Sự thừa kế (Inheritance) cho phép chúng ta được thừa hưởng những lớp đã có, giống như chúng ta có thể thừa hưởng những gì mà cha mẹ để lại, chúng ta cũng có thể tạo thêm những tính chất mới cho các lớp đã có để sử dụng. Trong C++, những lớp có trước gọi là lớp cơ sở, những lớp mở rộng gọi là lớp dẫn xuất.

✍ Sự đa hình (Polymorphism) bạn có thể hiểu một cách nôm na là chúng ta có thể gán một tên hoặc một kí hiệu cho một hàm (phương thức) và tên này có thể dùng cho toàn thế hệ nếu như hàm (phương thức) này có thể áp dụng được.

▪ Như các bạn đã biết ở trên sự đóng gói được thực hiện bằng một kiểu dữ liệu mới, đó là lớp. Vậy lớp là gì? Chúng tôi có thể nói với bạn như sau:

▪ Lớp (Class) thực chất là một kiểu dữ liệu được định nghĩa trong chương trình, tuy nhiên khái niệm về lớp còn đi xa hơn vì bên trong nó có sự cài đặt về các đặc tính cũng như các thao tác có thể có trên dữ liệu.

▪ Khi khai báo một Lớp ta dùng từ khoá **class tên_lớp**.

▪ Để hình dung và hiểu rõ về Lớp, bạn có thể xem qua ví dụ sau:

*Ví dụ 1:*

▪ Chúng ta tạo ra một lớp Haipheptoan để tính toán phép cộng và phép trừ.

▪ Trong C++ lớp này được định nghĩa như sau:

```
#include<iostream.h>
class haipheptoan
 {
   public :
   void cong();
   void trù ();
 };
```

```
void haipheptoan :: cong(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x+y;
        cout<<"x+y= "<<z;
    };
void haipheptoan :: tru(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x-y;
        cout<<"x-y= "<<z;
    };
    haipheptoan ketqua;
```

- Trong đoạn chương trình trên, chúng ta đã khai báo một lớp gồm các thành phần như sau:
- Từ khoá **Public** trong khai báo của lớp cho chúng ta biết những thành phần đứng sau Public được gọi là thành phần chung có thể được sử dụng bởi các hàm thành phần và trong toàn bộ chương trình.
- Kế đến là các hàm **void cong(void), void tru(void)**.
- Trong **class haipheptoan** chúng ta chỉ khai báo tên các hàm. Muốn xác định các hàm này, ta thực hiện theo cú pháp sau:

**Kiểu hàm Tên lớp: :Tên hàm (Kiểu biến)**

- Trong đoạn chương trình trên ta được:

```
void haipheptoan :: cong(void)
void haipheptoan :: tru(void)
```

Trong hàm voi haipheptoan :: cong (void)

+ Đầu tiên ta khai báo 3 biến x, y và z có kiểu nguyên.
+ Kế đến là nhập vào hai biến x và y.
+ Tính x+y cho vào biến z.
+ Xuất ra màn hình kết quả.

- Trong hàm voi haipheptoan :: tru (void)

+ Đầu tiên ta khai báo 3 biến x, y và z có kiểu nguyên.
+ Kế đến là nhập vào hai biến x và y.
+ Tính x-y cho vào biến z.
+ Xuất ra màn hình kết quả.

- Cuối cùng là khai báo biến Ketqua có kiểu haipheptoan qua câu lệnh

haipheptoan ketqua;

- Qua câu lệnh trên ta được ketqua là một lớp có các thành phần như lop haipheptoan.

*Ghi chú:*

- **Để khai báo lớp ta dùng từ khoá class tênlớp.**
- **Các dữ liệu trong lớp gọi là dữ liệu thành phần.**
- **Các hàm trong lớp gọi là hàm thành phần.**

- Để thực hiện đoạn chương trình trên bằng một kết quả cụ thể, bạn viết phải viết thêm một đoạn chương trình nữa, cụ thể bạn sẽ có chương trình như sau:

```
#include<iostream.h>
class haipheptoan
 {
   public :
   void cong();
   void tru ();
 };
 void haipheptoan :: cong(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
```

```
        z=x+y;
        cout<<"x+y= "<<z;
    };
void haipheptoan :: tru(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x-y;
        cout<<"x-y= "<<z;
    };

    haipheptoan ketqua;

main()
 {
   int banchon;
   do
    {
     cout<<"\n Ban chon 1, 2 hoac 3?"
          << "\n 1- Phep cong "
          << "\n 2- Phep tru "
          << "\n 3- Thoat ";
    cin>>banchon;
    switch(banchon)
        {
         case 1:
           {
                ketqua.cong();
                 break;
         }
         case 2 :\
           {
                ketqua.tru();
```

```
                    break;
                }
            case 3 : break;
            default : cout<<"Sai roi chon 1, 2, hoac 3";
        }
    } while(banchon !=3);
  return 0;
}
```

▪ Đoạn chương trình đậm ở trên là **đoạn phải thêm vào, trong hàm main( )**, đầu tiên là khai báo một biến nguyên có tên là banchon. Biến này dùng để ghi nhận kết quả khi bạn nhấn vào số 1, 2 hoặc 3 khi chạy chương trình.

▪ Vòng lặp **do ... while** để tạo một menu cho bạn chọn hai phép toán cộng hoặc trừ hoặc thoát khỏi chương trình tương ứng với các số 1, 2 hoặc 3 trên bàn phím.

▪ Khi chọn 1 hoặc 2, thực hiện xong, menu lại hiện ra, chỉ khi nào bạn chọn số 3 thì mới thoát khỏi chương trình.

▪ phát biểu **switch** sẽ chọn giá trị để thực hiện và thực hiện hàm nào.

▪ Nếu chạy thử chương trình trên, đầu tiên bạn sẽ thấy chương trình hiện ra menu với nội dung như sau:

**1- Phep cong**

**2- Phep tru**

**3- Thoat**

▪ Giả sử bạn chọn 1 bằng cách gõ phím số 1 rồi Enter. Bạn sẽ thấy trên màn hình câu

**Bạn nhập x =**

▪ Giả sử bạn nhập 6. Sau khi nhập xong, bạn sẽ thấy xuất hiện

**Bạn nhập y =**

▪ Giả sử bạn nhập 4. Sau khi nhập xong, bạn sẽ thấy

**x+y=10**

và xuất hiện lại menu để bạn thực hiện tiếp

**1- Phep cong**

**2- Phep tru**

**3- Thoat**

▪ Bạn cũng sẽ có quá trình tương tự nếu như bạn chọn số 2 và gõ Enter.

▪ Cứ tiếp tục thực hiện cho đến khi không muốn làm nữa, bạn chỉ cần bấm 3 trên bàn phím rồi Enter.

▪ Sau khi xem ví dụ này, bạn thấy lớp cũng không có gì phức tạp lắm phải không các bạn.

▪ Cũng ví dụ trên, nhưng bạn khai báo trực tiếp hàm trong class không cần phải khai báo tên hàm và dùng cú pháp để gọi hàm như trên. Xem ví dụ sau:

```
#include<iostream.h>
class haipheptoan
 {
   public :
   void cong(void)
     {
         int x;
         int y;
         int z;
         cout<<"Ban nhap x= ";
         cin>>x;
         cout<<"Ban nhap y= ";
         cin>>y;
         z=x+y;
         cout<<"x+y= "<<z;
     }
   void tru(void)
     {
         int x;
         int y;
         int z;
         cout<<"Ban nhap x= ";
         cin>>x;
         cout<<"Ban nhap y= ";
         cin>>y;
         z=x-y;
         cout<<"x-y= "<<z;
     }
 };
 haipheptoan ketqua;
```

```
main()
 {
   int banchon;
   do
    {
     cout<< "\n Ban chon 1, 2 hoac 3?"
           << "\n 1- Phep cong "
           << "\n 2- Phep tru "
           << "\n 3- Thoat ";
    cin>>banchon;
    switch(banchon)
        {
          case 1:
            {
                  ketqua.cong();
                   break;
            }
          case 2 :
            {
                  ketqua.tru();
                  break;
            }
          case 3 : break;
          default : cout<<"Sai roi chon 1, 2, hoac 3 ";
        }
 } while(banchon!=3);
 return 0;
}
```

• Qua chương trình này, bạn thấy chúng ta chỉ thay đổi một chút, nhìn có vẻ ngắn gọn hơn chương trình trước, theo bạn thì dùng cách nào hay hơn. Bạn cứ sử dụng theo ý thích của mình. Chạy thử chương trình này, các bạn thấy kết quả cũng giống như ở chương trình vừa rồi.

*Ví dụ 2:*

• Để hiểu rõ thêm về lớp, chúng tôi giới thiệu thêm cho các bạn một bài tập nữa, bài tập này các bạn sẽ tạo một lớp Thơ, trong lớp này khai báo hai hàm thành phần là **HTQ()** và **HMT()**. Hai hàm thành phần này

chứa thơ của Bà Huyện Thanh Quan và Hàn Mặc Tử, chương trình này được thể hiện như sau

```
#include<iostream.h>
class Tho
  {
    public :
    void HTQ();
    void HMT();
  };
void Tho :: HTQ(void)
  {
    cout<<"        QUA DEO NGANG ";
    cout<<"\n Buoc toi deo Ngang bong xe ta ";
    cout<<"\n Co cay chen da la chen hoa";
    cout<<"\n Lom khom duoi nui tieu vai chu";
    cout<<"\n Lac dac ben song ro may nha";
    cout<<"\n Nho nuoc dau long con quoc quoc ";
    cout<<"\n Thuong nha moi mieng cai gia gia";
    cout<<"\n Dung chan dung lai troi non nuoc ";
    cout<<"\n Mot manh tinh rieng ta voi ta";
  };

void Tho :: HMT(void)
  {
    cout<<"      DAY THON VI GIA ";
    cout<<"\n Sao anh khong ve choi thon Vi ";
    cout<<"\n Nhin nang hang cau nang moi len ";
    cout<<"\n Vuon ai muot qua xanh nhu ngoc ";
    cout<<"\n La truc che ngang mat chu dien ";
    cout<<"\n  ";
    cout<<"\n Gio theo loi gio may duong may ";
    cout<<"\n Dong nuoc buon thiu hoa bap lay ";
    cout<<"\n Thuyen ai doi ben song trang do ";
    cout<<"\n Co cho trang ve kip toi nay ";
    cout<<"\n  ";
    cout<<"\n Mo khach duong xa khach duong xa ";
    cout<<"\n Ao em trang qua nhin khong ra ";
    cout<<"\n O day suong khoi mo nhan anh ";
```

```
    cout<<"\n Moi biet tinh ai co dam da ";
  };
     Tho Thehien;
main()
  {
    int banchon;
    do
     {
      cout<<"\n Ban chon 1, 2 hoac 3?"
            << "\n 1- Tho Ba Huyen Thanh Quan "
            << "\n 2- Tho Han Mac Tu "
            << "\n 3- Thoat ";
      cin>>banchon;
      switch(banchon)
          {
            case 1:
              {
                    Thehien.HTQ();
                     break;
              }
            case 2 :
              {
                    Thehien.HMT();
                    break;
              }
            case 3 : break;
            default : cout<<"Sai roi chon 1, 2, hoac 3 ";
          }
  } while(banchon!=3);
  return 0;
}
```

▪ Chương trình này cũng giống như chương trình trên, bạn tự xem và hiểu nhé!

▪ Chạy thử chương trình trên, bạn thấy sẽ xuất hiện một menu như sau:

**Ban chon 1, 2 hoac 3?**

**1- Tho Ba Huyen Thanh Quan**

2- Tho Han Mac Tu
3- Thoat

▪ Nếu bạn nhập vào số 1, bạn sẽ thấy bài thơ Qua đèo Ngang của Bà Huyện Thanh Quan và menu của chương trình để bạn có thể chọn tiếp, bạn thấy như sau:

QUA DEO NGANG
Buoc toi deo Ngang bong xe ta
Co cay chen da la chen hoa
Lom khom duoi nui tieu vai chu
Lac dac ben song ro may nha
Nho nuoc dau long con quoc quoc
Thuong nha moi mieng cai gia gia
Dung chan dung lai troi non nuoc
Mot manh tinh rieng ta voi ta

Ban chon 1, 2 hoac 3?
1- Tho Ba Huyen Thanh Quan
2- Tho Han Mac Tu
3- Thoat

▪ Nếu chọn số 2 và gõ Enter, bạn sẽ thấy kết quả của chương trình báo cho bạn như sau:

DAY THON VI GIA
Sao anh khong ve choi thon Vi
Nhin nang hang cau nang moi len
Vuon ai muot qua xanh nhu ngoc
La truc che ngang mat chu dien

Gio theo loi gio may duong may
Dong nuoc buon thiu hoa bap lay
Thuyen ai doi ben song trang do
Co cho trang ve kip toi nay

Mo khach duong xa khach duong xa

**Ao em trang qua nhin khong ra**
**O day suong khoi mo nhan anh**
**Moi biet tinh ai co dam da**

**Ban chon 1, 2 hoac 3?**
**1- Tho Ba Huyen Thanh Quan**
**2- Tho Han Mac Tu**
**3- Thoat**

- Nếu gõ vào số 3 và nhấn phím Enter, bạn sẽ thoát khỏi chương trình.
- Vậy là phần đóng gói qua sự thể hiện của lớp chắc bạn đã hiểu rồi phải không, bây giờ chúng tôi sẽ nói qua phần **sự thừa kế**, như các bạn đã biết ở trên, sự thừa kế từ **một lớp đã có gọi là lớp cơ sở, lớp mở rộng sau này gọi là lớp dẫn xuất**
- Chúng tôi sẽ dùng hai ví dụ trên để minh họa cho sự thừa kế.
- Trong ví dụ thứ nhất ở trên, các bạn dùng một lớp haipheptoan, bây giờ từ lớp đã có này, chúng tôi sẽ thừa kế lớp này, chúng tôi sẽ thêm hai phép toán nữa, đó là phép toán nhân và phép toán chia, bạn sẽ tham khảo qua đoạn chương trình sau

```
#include<iostream.h>
class haipheptoan
 {
   public :
   void cong();
   void tru();
 };
 class them : public haipheptoan
  {
    public :
      void nhan();
      void chia();
  }bonpheptoan;
 void haipheptoan :: cong(void)
  {
        int x;
        int y;
```

```
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x+y;
        cout<<"x+y= "<<z;
  };
 void haipheptoan :: tru(void)
  {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x-y;
        cout<<"x-y= "<<z;
  };

 void them :: nhan(void)
  {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x*y;
        cout<<"x*y= "<<z;
  };
void them :: chia(void)
  {
        int x;
        int y;
        int z;
```

```
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y Khac 0, y= ";
        cin>>y;
        z=x/y;
        cout<<"x/y= "<<z;
  };
main()
{
   int banchon;
   do
    {
     cout<<"\n Ban chon 1, 2, 3, 4 hoac 5?"
          << "\n 1- Phep cong "
          << "\n 2- Phep tru "
          << "\n 3- Phep nhan "
          << "\n 4- Phep chia "
          << "\n 5- Thoat " ;
    cin >> banchon;
    switch(banchon)
        {
         case 1:
           {
                bonpheptoan.cong();
                break;
           }
         case 2 :
           {
                bonpheptoan.tru();
                break;
           }
         case 3:
           {
                bonpheptoan.nhan();
                break;
           }
         case 4 :
           {
```

```
                bonpheptoan.chia();
                break;
            }
        case 5 : break;
        default : cout<<"Sai roi chon 1, 2, 3, 4 hoac 5 ";
    }
  } while(banchon!=5);
  return 0;
}
```

▪ Qua chương trình này, chúng tôi tạo thêm một lớp mới có tên là Them, lớp này thừa kế lớp Haipheptoan, trong lớp Them này, chúng tôi khai báo thêm hai phép toán nữa đó là phép toán nhân và phép toán chia.

▪ Để tạo một lớp thừa kế một lớp nào đó, bạn dùng cú pháp như sau :

**Lớp mới (lớp dẫn xuất):Lớp đã có (lớp cơ sở)**

▪ Trong chương trình trên, chúng ta dùng:

**class them: public haipheptoan**

▪ Và chúng ta đặt đối tượng Bonpheptoan có kiểu them. Và ta sẽ dùng đối tượng này để thực hiện bốn phép toán trong chương trình.

- ❖ Lớp Haipheptoan là lớp cơ sở gồm hai hàm thành phần đó là Cong(), Tru().
- ❖ Lớp them được mở rộng thêm hai hàm nữa, đó là Nhan() và Chia()
- ❖ Cuối cùng lớp Them xem như có bốn phép toán cộng, trừ, nhân và chia .

▪ Đối tượng Bonpheptoan có kiểu them sẽ được dùng trong chương trình, bạn thấy ở Code bên trên.

▪ Chạy thử chương trình, bạn sẽ thấy xuất hiện một menu như sau:

```
Ban chon 1, 2, 3, 4 hoac 5?"
        1- Phep cong "
        2- Phep tru "
        3- Phep nhan "
        4- Phep chia "
        5- Thoat " ;
```

▪ Nếu chọn số 1 và gõ Enter, bạn sẽ thực hiện phép cộng hai số, nếu chọn số 2 thì thực hiện phép trừ ... Nếu muốn thoát khỏi chương trình, bạn chọn số 5 và gõ Enter.

▪ Trong các lần bạn chọn, chương trình sẽ cho bạn nhập x, nhập y, và sau đó sẽ cho kết quả. Bạn chạy thử và sẽ thấy được kết quả như chúng tôi đã nói ở trên.

▪ Chúng tôi giới thiệu đến các bạn một ví dụ nữa qua ví dụ 2 ở trên.

▪ Trong ví dụ này, chúng tôi sẽ thừa kế lớp đã có đó là lớp Tho, trong lớp này có hai hàm đó là HTQ() và hàm HMT(), lớp mở rộng là lớp Them thừa kế lớp Tho ở trên có thêm hai hàm đó là TX() và NB(0).

▪ Đối tượng Thotonghop có kiểu Them tác động đến bốn hàm như ở chương trình sau:

```
#include<iostream.h>
class Tho
 {
   public :
   void HTQ();
   void HMT ();
 };
 void Tho :: HTQ(void)
    {
        cout<<"        QUA DEO NGANG ";
        cout<<"\n Buoc toi deo Ngang bong xe ta ";
        cout<<"\n Co cay chen da la chen hoa";
        cout<<"\n Lom khom duoi nui tieu vai chu";
        cout<<"\n Lac dac ben song ro may nha";
        cout<<"\n Nho nuoc dau long con quoc quoc ";
        cout<<"\n Thuong nha moi mieng cai gia gia";
        cout<<"\n Dung chan dung lai troi non nuoc ";
        cout<<"\n Mot manh tinh rieng ta voi ta";
    };
 void Tho :: HMT(void)
    {
        cout<<"      DAY THON VI GIA ";
        cout<<"\n Sao anh khong ve choi thon Vi ";
        cout<<"\n Nhin nang hang cau nang moi len ";
        cout<<"\n Vuon ai muot qua xanh nhu ngoc ";
        cout<<"\n La truc che ngang mat chu dien ";
        cout<<"\n  ";
        cout<<"\n Gio theo loi gio may duong may ";
        cout<<"\n Dong nuoc buon thiu hoa bap lay ";
```

```
        cout<<"\n Thuyen ai doi ben song trang do ";
        cout<<"\n Co cho trang ve kip toi nay ";
        cout<<"\n  ";
        cout<<"\n Mo khach duong xa khach duong xa ";
        cout<<"\n Ao em trang qua nhin khong ra ";
        cout<<"\n O day suong khoi mo nhan anh ";
        cout<<"\n Moi biet tinh ai co dam da ";
    };
class Them : public Tho
    {
        public :
        void TX();
        void NB();
    }Thotonghop;
  void Them :: TX(void)
    {
       cout<<"          THOI DOI ";
       cout<<"\n Nguoi bao ong dien ong chang dien ";
       cout<<"\n Ong thuong ong tiec hoa ong phien ";
       cout<<"\n Ke yeu nguoi ghet hay thi chu ";
       cout<<"\n Dua trong thang khinh chi vi tien ";
       cout<<"\n O be ngam ngui con toi lach ";
       cout<<"\n Duoc voi tap tenh lai doi tien ";
       cout<<"\n Khi cuoi khi khoc khi than tho ";
       cout<<"\n Muon bo van chuong hoc vo bien ";
    };
 void Them :: NB(void)
   {
       cout<<"          LONG ME ";
       cout<<"\n Gai lon ai khong phai lay chong ";
       cout<<"\n Can gi ma khoc nin di khong ";
       cout<<"\n Nin di, mac ao ra chao me ";
       cout<<"\n Ro quy con toi cac chi trong ";
       cout<<"\n  ";
       cout<<"\n Uong uong do do qua di thoi ";
       cout<<"\n Co co con thuong den chung toi ";
       cout<<"\n Thi dung len nao lau nuoc mat ";
       cout<<"\n Minh co lam ban may muoi nguoi ";
```

```
        cout<<"\n  ";
        cout<<"\n Nay ao dong lam quan lanh tia ";
        cout<<"\n Nay guong nay luoc nay hoa tai ";
        cout<<"\n Muon gi toi sam cho co du ";
        cout<<"\n Nao da thua ai da kem ai ";
        cout<<"\n  ";
        cout<<"\n Ruong toi cay cay dau toi hai ";
        cout<<"\n Nuoi day em co toi dam duong ";
        cout<<"\n Nha cua toi coi, no toi tra ";
        cout<<"\n Toi con manh chan, khien co thuong ";
        cout<<"\n  ";
        cout<<"\n Dua con ra den cua buong thoi ";
        cout<<"\n Me phai xa con kho may muoi ";
        cout<<"\n Con a! dem nay minh me khoc ";
        cout<<"\n Dem dem minh me lai dua thoi ";
    };
main()
  {
    int banchon;
    do
     {
      cout << "\n Ban chon 1 , 2, 3, 4 hoac 5 ? "
           << "\n 1- Tho Ba Huyen Thanh Quan "
           << "\n 2- Tho Han Mac Tu "
           << "\n 3- Tho Tu Xuong "
           << "\n 4- Tho Nguyen Binh"
           << "\n 5- Thoat ";
      cin >> banchon;
      switch(banchon)
          {
           case 1:
             {
                 Thotonghop.HTQ();
                  break;
             }
           case 2 :
             {
```

```
                Thotonghop.HMT();
                break;
            }
        case 3:
            {
                Thotonghop.TX();
                 break;
            }
        case 4 :
            {
                Thotonghop.NB();
                break;
            }
        case 5 : break;
        default : cout<< " Sai roi chon 1, 2, 3, 4 hoac 5 ";
      }
  } while(banchon!=5);
  return 0;
}
```

▪ Nếu chạy thử chương trình, bạn cũng thấy khá giống như ở chương trình trước, menu chỉ khác ở chỗ thay các phép toán bằng các bài thơ.

▪ Bạn cũng có thể thấy được là từ **một lớp có thể thừa kế đồng thời từ hai lớp đã có**, có nghĩa là **một lớp có thể là một lớp dẫn xuất của nhiều lớp cơ sở**. Trong ví dụ sau đây, bạn sẽ thấy

- Lớp cơ sở thứ nhất là haipheptoan, có hai hàm thành phần là Cong() và Tru().
- Lớp cơ sở thứ hai là Tho, có hai hàm thành phần là HTQ() và HMT().
- Bạn tạo một lớp dẫn xuất có tên là Danxuat thừa kế hai lớp cơ sở ở trên và có thêm bốn hàm là Nhan(), Chia(), TX() và NB().
- Cho đối tượng kethop có kiểu là danxuat.

▪ Ví dụ sau sẽ minh hoạ những điều vừa nói, chúng tôi kết hợp các ví dụ ở trên để giới thiệu cùng các bạn.

```
#include<iostream.h>
class haipheptoan
 {
   public :
   void cong();
```

```
    void tru ();
  };
void haipheptoan :: cong(void)
      {
          int x;
          int y;
          int z;
          cout<<"Ban nhap x= ";
          cin>>x;
          cout<<"Ban nhap y= ";
          cin>>y;
          z=x+y;
          cout<<"x+y= "<<z;
      };
  void haipheptoan :: tru(void)
      {
          int x;
          int y;
          int z;
          cout<<" Ban nhap x= ";
          cin>>x;
          cout<<"Ban nhap y= ";
          cin>>y;
          z=x-y;
          cout<<"x-y= "<<z;
      };
class Tho
  {
    public :
    void HTQ();
    void HMT();
  };
void Tho :: HTQ(void)
      {
          cout<<"        QUA DEO NGANG ";
          cout<<"\n Buoc toi deo Ngang bong xe ta ";
          cout<<"\n Co cay chen da la chen hoa";
          cout<<"\n Lom khom duoi nui tieu vai chu";
```

```
        cout<<"\n Lac dac ben song ro may nha";
        cout<<"\n Nho nuoc dau long con quoc quoc ";
        cout<<"\n Thuong nha moi mieng cai gia gia";
        cout<<"\n Dung chan dung lai troi non nuoc ";
        cout<<"\n Mot manh tinh rieng ta voi ta";
    };
void Tho :: HMT(void)
    {
        cout<<"      DAY THON VI GIA ";
        cout<<"\n Sao anh khong ve choi thon Vi ";
        cout<<"\n Nhin nang hang cau nang moi len ";
        cout<<"\n Vuon ai muot qua xanh nhu ngoc ";
        cout<<"\n La truc che ngang mat chu dien ";
        cout<<"\n  ";
        cout<<"\n Gio theo loi gio may duong may ";
        cout<<"\n Dong nuoc buon thiu hoa bap lay ";
        cout<<"\n Thuyen ai doi ben song trang do ";
        cout<<"\n Co cho trang ve kip toi nay ";
        cout<<"\n  ";
        cout<<"\n Mo khach duong xa khach duong xa ";
        cout<<"\n Ao em trang qua nhin khong ra ";
        cout<<"\n O day suong khoi mo nhan anh ";
        cout<<"\n Moi biet tinh ai co dam da ";
    };
class danxuat : public haipheptoan, public Tho
  {
    public :
        void nhan(void);
        void chia(void);
        void TX(void);
        void NB(void);
  }kethop;
void danxuat :: TX(void)
  {
      cout<<"          THOI DOI ";
      cout<<"\n Nguoi bao ong dien ong chang dien ";
      cout<<"\n Ong thuong ong,tiec hoa ong phien ";
      cout<<"\n Ke yeu nguoi ghet hay thi chu ";
```

```
        cout<<"\n Dua trong thang khinh chi vi tien ";
        cout<<"\n O be ngam ngui con toi lach ";
        cout<<"\n Duoc voi tap tenh lai doi tien ";
        cout<<"\n Khi cuoi khi khoc khi than tho ";
        cout<<"\n Muon bo van chuong hoc vo bien ";
    };
void danxuat :: NB(void)
    {
        cout<<"        LONG ME ";
        cout<<"\n Gai lon ai khong phai lay chong ";
        cout<<"\n Can gi ma khoc nin di khong ";
        cout<<"\n Nin di, mac ao ra chao me ";
        cout<<"\n Ro quy con toi cac chi trong ";
        cout<<"\n  ";
        cout<<"\n Uong uong do do qua di thoi ";
        cout<<"\n Co co con thuong den chung toi ";
        cout<<"\n Thi dung len nao lau nuoc mat ";
        cout<<"\n Minh co lam ban may muoi nguoi ";
        cout<<"\n  ";
        cout<<"\n Nay ao dong lam quan lanh tia ";
        cout<<"\n Nay guong nay luoc nay hoa tai ";
        cout<<"\n Muon gi toi sam cho co du ";
        cout<<"\n Nao da thua ai da kem ai ";
        cout<<"\n  ";
        cout<<"\n Ruong toi cay cay dau toi hai ";
        cout<<"\n Nuoi day em co toi dam duong ";
        cout<<"\n Nha cua toi coi, no toi tra ";
        cout<<"\n Toi con manh chan, khien co thuong ";
        cout<<"\n  ";
        cout<<"\n Dua con ra den cua buong thoi ";
        cout<<"\n Me phai xa con kho may muoi ";
        cout<<"\n Con a! dem nay minh me khoc ";
        cout<<"\n Dem dem minh me lai dua thoi ";
    };
void danxuat :: nhan(void)
    {
        int x;
        int y;
```

```
        int z;
        cout<<" Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x*y;
        cout<<"x*y= "<<z;
    };
  void danxuat :: chia(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y Khac 0, y= ";
        cin>>y;
        z=x/y;
        cout<<"x/y= "<<z;
    };
main()
  {
    int banchon;
    do
     {
      cout << "\n Ban chon 1, 2, 3, 4, 5, 6, 7, 8 hoac 9?"
             << "\n 1- Phep cong "
             << "\n 2- Phep tru "
             << "\n 3- Phep nhan "
             << "\n 4- Phep chia "
             << "\n 5- Tho Ba Huyen Thanh Quan "
             << "\n 6- Tho Han Mac Tu "
             << "\n 7- Tho Tu Xuong "
             << "\n 8- Tho Nguyen Binh "
             << "\n 9- Thoat " ;
     cin>>banchon;
     switch(banchon)
```

```
{
  case 1:
      {
            kethop.cong();
            break;
      }
  case 2:
      {
            kethop.tru();
            break;
      }
  case 3:
      {
            kethop.nhan();
            break;
      }
  case 4 :
      {
            kethop.chia();
            break;
      }
  case 5:
      {
            kethop.HTQ();
            break;
      }
  case 6 :
      {
            kethop.HMT();
            break;
      }
  case 7 :
      {
            kethop.TX();
            break;
      }
  case 8 :
      {
```

```
                kethop.NB();
                break;
            }
        case 9 : break;
          default : cout<<"Sai roi chon 1, 2, 3, 4, 5, 6, 7, 8 hoac 9 ";
        }
  } while(banchon!=9);
 return 0;
}
```

- Qua chương trình trên, bạn thấy bạn có thể tạo một lớp cùng thừa kế từ hai lớp đã có, vậy bạn có thể tạo một lớp thừa kế nhiều lớp mới.
- Trong chương trình này, hai lớp đã có trước đó là
  - ❖ Lớp Haipheptoan
  - ❖ Lớp Tho

    Trong lớp Haipheptoan có hai hàm thành phần đó là Cong() và Trừ().

    Trong lớp Tho sẽ có hai hàm thành phần đó là HTQ() và HMT().

Sau đó các bạn thấy chúng tôi tạo ra lớp có tên Danxuat thừa kế hai lớp cơ sở ở trên. Trong lớp dẫn xuất này, chúng tôi tạo thêm bốn hàm nữa, đó là TX(), NB(), Nhan() và Chia(). Bây giờ xem như trong lớp Danxuat sẽ có tất cả là 8 hàm, đó là:

Cong()
Tru()
Nhan()
Chia()
HTQ()
HMT()
TX()
NB()

Đối tượng kethop có kiểu là Danxuat, dùng đối tượng này để thực hiện trong chương trình trên.

Nếu chạy thử chương trình trên, bạn sẽ có menu như sau:

```
 Ban chon 1, 2, 3, 4, 5, 6, 7, 8 hoac 9
            1- Phep cong
            2- Phep tru
```

3- Phep nhan
4- Phep chia
5- Tho Ba Huyen Thanh Quan
6- Tho Han Mac Tu
7- Tho Tu Xuong
8- Tho Nguyen Binh
9- Thoat

▪ Khi nhập một số trong các số từ 1 đến 9, chương trình sẽ thực hiện theo các ghi chú tương ứng. Bạn tự kiểm tra nhé!

▪ Tiếp theo bạn xem một ví dụ để thấy **một lớp dẫn xuất của một lớp cơ sở, có thể là lớp cơ sở của lớp dẫn xuất mới**

```
#include<iostream.h>
class haipheptoan
 {
   public :
   void cong();
   void tru ();
 };
 void haipheptoan :: cong(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x+y;
        cout<<"x+y= "<<z;
    };
 void haipheptoan :: tru(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
```

```
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x-y;
        cout<<"x-y= "<<z;
    };
class Thovatoan: public haipheptoan
 {
   public :
   void HTQ();
   void HMT();
   void TX();
   void NB();
 };
 void Thovatoan :: HTQ(void)
    {
        cout<<"        QUA DEO NGANG ";
        cout<<"\n Buoc toi deo Ngang bong xe ta ";
        cout<<"\n Co cay chen da la chen hoa";
        cout<<"\n Lom khom duoi nui tieu vai chu";
        cout<<"\n Lac dac ben song ro may nha";
        cout<<"\n Nho nuoc dau long con quoc quoc ";
        cout<<"\n Thuong nha moi mieng cai gia gia";
        cout<<"\n Dung chan dung lai troi non nuoc ";
        cout<<"\n Mot manh tinh rieng ta voi ta";
    };
 void Thovatoan :: HMT(void)
    {
        cout<<"      DAY THON VI GIA ";
        cout<<"\n Sao anh khong ve choi thon Vi ";
        cout<<"\n Nhin nang hang cau nang moi len ";
        cout<<"\n Vuon ai muot qua xanh nhu ngoc ";
        cout<<"\n La truc che ngang mat chu dien ";
        cout<<"\n  ";
        cout<<"\n Gio theo loi gio may duong may ";
        cout<<"\n Dong nuoc buon thiu hoa bap lay ";
        cout<<"\n Thuyen ai doi ben song trang do ";
        cout<<"\n Co cho trang ve kip toi nay ";
        cout<<"\n  ";
```

```
        cout<<"\n Mo khach duong xa khach duong xa ";
        cout<<"\n Ao em trang qua nhin khong ra ";
        cout<<"\n O day suong khoi mo nhan anh ";
        cout<<"\n Moi biet tinh ai co dam da ";
  };
void Thovatoan :: TX(void)
 {
        cout<<"          THOI DOI ";
        cout<<"\n Nguoi bao ong dien ong chang dien ";
        cout<<"\n Ong thuong ong tiec hoa ong phien ";
        cout<<"\n Ke yeu nguoi ghet hay thi chu ";
        cout<<"\n Dua trong thang khinh chi vi tien ";
        cout<<"\n O be ngam ngui con toi lach ";
        cout<<"\n Duoc voi tap tenh lai doi tien ";
        cout<<"\n Khi cuoi khi khoc khi than tho ";
        cout<<"\n Muon bo van chuong hoc vo bien ";
  };
void Thovatoan :: NB(void)
 {
        cout<<"        LONG ME ";
        cout<<"\n Gai lon ai khong phai lay chong ";
        cout<<"\n Can gi ma khoc nin di khong ";
        cout<<"\n Nin di, mac ao ra chao me ";
        cout<<"\n Ro quy con toi cac chi trong ";
        cout<<"\n  ";
        cout<<"\n Uong uong do do qua di thoi ";
        cout<<"\n Co co con thuong den chung toi ";
        cout<<"\n Thi dung len nao lau nuoc mat ";
        cout<<"\n Minh co lam ban may muoi nguoi ";
        cout<<"\n  ";
        cout<<"\n Nay ao dong lam quan lanh tia ";
        cout<<"\n Nay guong nay luoc nay hoa tai ";
        cout<<"\n Muon gi toi sam cho co du ";
        cout<<"\n Nao da thua ai da kem ai ";
        cout<<"\n  ";
        cout<<"\n Ruong toi cay cay dau toi hai ";
        cout<<"\n Nuoi day em co toi dam duong ";
        cout<<"\n Nha cua toi coi, no toi tra ";
```

```
        cout<<"\n Toi con manh chan , khien co thuong ";
        cout<<"\n  ";
        cout<<"\n Dua con ra den cua buong thoi ";
        cout<<"\n Me phai xa con kho may muoi ";
        cout<<"\n Con ai dem nay minh me khoc ";
        cout<<"\n Dem dem minh me lai dua thoi ";
    };
class Lopcuoi:public Thovatoan
  {
    public:
    void nhan();
    void chia();
  }kethop;
  void Lopcuoi :: nhan(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y= ";
        cin>>y;
        z=x*y;
        cout<<"x*y= "<<z;
    };

   void Lopcuoi :: chia(void)
    {
        int x;
        int y;
        int z;
        cout<<"Ban nhap x= ";
        cin>>x;
        cout<<"Ban nhap y khac 0, y= ";
        cin>>y;
        z=x/y;
        cout<<"x/y= "<<z;
    };
```

```
main()
  {
    int banchon;
    do
     {
      cout << "\n Ban chon 1, 2, 3, 4, 5, 6, 7, 8 hoac 9? "
             << "\n 1- Phep cong "
             << "\n 2- Phep tru "
             << "\n 3- Phep nhan "
             << "\n 4- Phep chia "
             << "\n 5- Tho Ba Huyen Thanh Quan "
             << "\n 6- Tho Han Mac Tu "
             << "\n 7- Tho Tu Xuong "
             << "\n 8- Tho Nguyen Binh "
             << "\n 9- Thoat " ;
      cin>>banchon;
switch(banchon)
      {
           case 1:
               {
                    kethop.cong();
                    break;
               }
          case 2 :
               {
                    kethop.tru();
                    break;
               }
         case 3:
               {
                    kethop.nhan();
                    break;
               }
         case 4 :
               {
                    kethop.chia();
                    break;
               }
```

```
        case 5:
            {
                kethop.HTQ();
                break;
            }
        case 6 :
            {
                kethop.HMT();
                break;
            }
        case 7:
            {
                kethop.TX();
                break;
            }
        case 8 :
            {
                kethop.NB();
                break;
            }
        case 9 : break;
        default : cout<< " Sai roi chon 1, 2, 3, 4, 5, 6, 7, 8 hoac 9 ";
    }
} while(banchon!=9);
  return 0;
  }
```

- Trong chương trình trên, kết quả khi thực hiện cũng giống như chương trình trước, từ Menu của chương trình cho đến kết quả thể hiện. Nó chỉ khác cung cách lập trình.
- Ở bài này, đầu tiên từ lớp cơ sở có sẵn là lớp Haipheptoan, chúng tôi tạo một lớp có tên là Thovatoan, lớp này thừa kế lớp Haipheptoan. Vậy lớp Thovatoan là lớp dẫn xuất của lớp haipheptoan.
- Sau đó chúng tôi tạo thêm một lớp mới có tên là Tonghop, lớp này sẽ thừa kế lớp Thovatoan. Vậy các bạn thấy

❖ Lớp Thovatoan sẽ đóng vai trò là lớp dẫn xuất đối với lớp Haipheptoan.
❖ Lớp Thovatoan sẽ đóng vai trò là lớp cơ sở của lớp dẫn xuất Tonghop.

▪ Vậy ta có thể tạo theo tính cách dây chuyền như trên, để cuối cùng ta được một lớp sẽ chứa tất cả các hàm của các lớp trên. Đây là một tính ưu việt trong lập trình hướng đối tượng, nhờ vào các lớp đã có, nếu chúng ta có thể sử dụng được trong chương trình, hoặc ta có thể dùng một lớp đã có để tạo ra một lớp mới phong phú hơn. Trong Window cũng dùng như vậy, nên các bạn thấy các ngôn ngữ ngày càng mạnh, bởi vì chỉ cần thêm một số điều gì đó, ta sẽ có cái mới hơn... Hy vọng những kiến thức này sẽ giúp các bạn sáng tạo ra những điều hay hơn từ những điều đã có trước .

▪ Sau đây chúng tôi giới thiệu đến các bạn về sự tác động của các **Hàm (Phương thức) thiết lập** và **Hàm (Phương thức) hủy bỏ**

▪ Phương thức thiết lập là một **phương thức thành phần (Hàm thành phần) của một lớp** và nó có cùng tên với lớp này. Ta có thể hình dung phương thức thiết lập để cấp phát ký ức và gán giá trị cho các dữ liệu thành phần.

▪ Phương thức hủy bỏ là một **phương thức thành phần (Hàm thành phần) của một lớp,** nó có cùng tên với lớp này và có dấu ngã "~" đứng trước. Phương thức này có tác dụng ngược lại với phương thức thiết lập. Ta có thể hình dung phương thức huỷ bỏ để xoá bỏ một đối tượng khi đối tượng đó không cần thiết nữa.

▪ Ta nên biết thêm điều này, khi lớp dẫn xuất và thừa kế đều có phương thức thiết lập chuẩn và phương thức huỷ bỏ, thứ tự thực hiện của chúng như trình tự cấp phát và thu hồi bộ nhớ.

▪ Bạn có thể xem các ví dụ sau để có thể thấy được trình tự làm việc của phương thức thiết lập và phương thức huỷ bỏ. Chúng tôi có thể nói ngay cho các bạn hiểu một cách nôm na như sau.

▪ Một người cấp phát cho bạn một thửa ruộng, bạn chia ra hai trường hợp sau:

◎ Nếu như vừa cấp phát xong, bạn chưa làm được lợi ích gì trên thửa ruộng đó, người chủ lấy lại, bạn nghĩ sao về điều này. Suy luận trong chương trình, giả sử hai biến a và b vừa được cấp phát bộ nhớ, chưa sử dụng gì cả, máy thu hồi lại ngay -> chương trình chưa thực hiện được.

◎ Nếu cấp phát cho bạn thửa ruộng, sau một thời gian hoàn thành công việc, bạn không cần sử dụng nữa, người chủ thu hồi lại và có thể phân phát cho người khác. Cũng vậy trong chương trình, với sự

cấp phát bộ nhớ cho hai biến a và b, sau khi chương trình sử dụng xong, máy sẽ lấy lại ngay để khỏi tốn bộ nhớ

*Ví dụ minh hoạ 1:*

- Trong ví dụ này, chúng tôi sẽ dùng phương thức thiết lập và phương thức hủy bỏ với mục đích muốn cho các bạn thấy được trình tự thực hiện của chúng. Bạn xem nội dung của chương trình sau:

```
#include<iostream.h>
class chucmung1
 {
   public :
     chucmung1()
        {
          cout<<"Chuc mung ban mot nam moi an khang \n";
        }
    ~chucmung1()
        {
          cout<< "Nam TAN TY \n";
        }
 };
 class chucmung2 : public chucmung1
 {
    public :
        chucmung2()
         {
           cout<<"Chuc mung ban mot nam moi truong tho \n";
         }
       ~chucmung2()
         {
           cout<< "Nam NHAM NGO \n";
         }
};
class chucmung3 : public chucmung2
{
   public :
       chucmung3()
         {
           cout<<"Chuc mung ban mot nam moi Hanh phuc \n";
         }
```

```
        ~chucmung3()
          {
            cout<< "Nam QUY MUI \n";
          }
};
class chucmung4 : public chucmung3
 {
   public :
      chucmung4()
          {
              cout<<"AN KHANG TRUONG THO VA HANH PHUC \n";
          }
       ~chucmung4()
          {
            cout<< " TAN TY - NHAM NGO - QUY MUI \n";
          }
};
void main()
  {
      chucmung4 object;
  }
```

- Đối với chương trình này, đầu tiên chúng tôi xây dựng một lớp có tên là Chucmung1 (chúc mừng1). Trong lớp này, chúng tôi xây dựng một phương thức thiết lập có tên là Chucmung1 (giống tên của lớp, điều này chúng tôi đã giới thiệu ở phần lý thuyết), và phương thức thiết lập này cũng là phương thức thành phần của lớp chucmung1.

- Trong phương thức thiết lập này, chúng tôi dùng một dòng xuất để xuất ra màn hình câu "Chúc mừng bạn một năm mới an khang" thể hiện như sau:

```
chucmung1( )
  {
    cout<<"Chuc mung ban mot nam moi an khang \n";
  }
```

- Trong lớp chucmung1 này chúng tôi cũng xây dựng một phương thức huỷ bỏ, phương thức này cũng là một phương thức thành phần của lớp chucmung1, chúng tôi dùng dấu ~ đứng trước tên của phương thức.

▪ Phương thức này, chúng tôi muốn xuất ra trên màn hình câu "Nam TAN TY", nó được viết như sau:

```
~chucmung1()
        {
          cout<< "Nam TAN TY \n";
        }
```

▪ Tiếp đến chúng tôi tạo một lớp dẫn xuất mới có tên chucmung2 thừa kế lớp cơ sở chucmung1 bằng câu lệnh

class chucmung2 : public chucmung1

▪ Câu lệnh trên bạn đã biết trong phần lý thuyết ở trên. Trong lớp dẫn xuất mới này, chúng tôi cũng xây dựng hai phương thức là phương thức thiết lập và phương thức hủy bỏ, trước hết chúng tôi xây dựng một lớp có tên là Chucmung2 (chúc mừng 2). Trong lớp này, chúng tôi xây dựng một phương thức thiết lập có tên là Chucmung2 (giống tên của lớp mới này), và phương thức thiết lập này cũng là phương thức thành phần của lớp chucmung2.

▪ Trong phương thức thiết lập này, chúng tôi dùng một dòng xuất để xuất ra màn hình câu "Chúc mừng bạn một năm mới trường thọ" thể hiện như sau:

```
Chucmung2( )
  {
    cout<<"Chuc mung ban mot nam moi truong tho\n";
  }
```

▪ Lớp chucmung2 này, chúng tôi cũng xây dựng một phương thức huỷ bỏ, phương thức này cũng là một phương thức thành phần của lớp chucmung2, chúng tôi dùng dấu ~ đứng trước tên của phương thức.

▪ Với phương thức này, chúng tôi muốn xuất ra trên màn hình câu "Nam NHAM NGO". Nó được viết như sau:

```
~chucmung2()
        {
          cout<< "Nam NHAM NGO \n";
        }
```

▪ Tiếp đến chúng tôi tạo một lớp dẫn xuất mới có tên chucmung3 thừa kế lớp cơ sở chucmung2 bằng câu lệnh

class chucmung3 : public chucmung2

▪ Trong lớp dẫn xuất mới này, chúng tôi cũng xây dựng hai phương thức là phương thức thiết lập và phương thức hủy bỏ, trước hết chúng tôi xây dựng một lớp có tên là Chucmung3 (chúc mừng 3). Trong lớp này, chúng tôi xây dựng một phương thức thiết lập có tên là Chucmung3 (giống tên của lớp mới này), và phương thức thiết lập này cũng là phương thức thành phần của lớp chucmung3.

▪ Với phương thức thiết lập này, chúng tôi dùng một dòng xuất để xuất ra màn hình câu "Chúc mừng bạn một năm mới Hạnh phúc" thể hiện như sau:

```
Chucmung3( )
  {
    cout<<"Chuc mung ban mot nam moi Hanh phuc\n";
  }
```

▪ Trong lớp chucmung3 này, chúng tôi cũng xây dựng một phương thức huỷ bỏ, phương thức này cũng là một phương thức thành phần của lớp chucmung3, chúng tôi dùng dấu ~ đứng trước tên của phương thức.

▪ Với phương thức này, chúng tôi muốn xuất ra trên màn hình câu "Nam QUY MUI". Nó được viết như sau:

```
~chucmung3()
      {
        cout<<"Nam QUY MUI \n";
      }
```

▪ Tiếp đến chúng tôi tạo một lớp dẫn xuất mới có tên chucmung4 thừa kế lớp cơ sở chucmung3 bằng câu lệnh

```
class chucmung4 : public chucmung3
```

▪ Trong lớp dẫn xuất mới này, chúng tôi cũng xây dựng hai phương thức đó là phương thức thiết lập và phương thức hủy bỏ, trước hết chúng tôi xây dựng một lớp có tên là Chucmung4 (chúc mừng 4). Trong lớp này, chúng tôi xây dựng một phương thức thiết lập có tên là Chucmung4 (giống tên của lớp mới này), và phương thức thiết lập này cũng là phương thức thành phần của lớp chucmung4.

▪ Với phương thức thiết lập này, chúng tôi dùng một dòng xuất để xuất ra màn hình câu "AN KHANG - TRUONG THO - HANH PHUC" thể hiện như sau:

```
Chucmung4( )
  {
```

```
    cout<<"AN KHANG - TRUONG THO - HANH PHUC\n";
}
```

▪ Trong lớp chucmung4 này, chúng tôi cũng xây dựng một phương thức huỷ bỏ, phương thức này cũng là một phương thức thành phần của lớp chucmung4, chúng tôi dùng dấu ~ đứng trước tên của phương thức.

▪ Với phương thức này, chúng tôi muốn xuất ra màn hình câu "TAN TY - NHAM NGO - QUY MUI". Nó được viết như sau:

```
~chucmung3()
    {
        cout<<"TAN TY - NHAM NGO - QUY MUI \n";
    }
```

▪ Theo bạn thì các dòng hiện ra sẽ như thế nào, bạn thử đoán trước khi chạy thử chương trình. Bạn đoán xong chưa, bây giờ bạn thử xem chương trình chạy như thế nào nhé. Kết quả sẽ như sau:

```
1-  Chúc mừng bạn một năm mới an khang
2-  Chúc mừng bạn một năm mới trường thọ
3-  Chúc mừng bạn một năm mới hạnh phúc
4-  AN KHANG - TRƯỜNG THỌ - HẠNH PHÚC
5-  TÂN TỴ - NHÂM NGỌ - QUÝ MUI
6-  Năm QUÝ MÙI
7-  Năm NHÂM NGỌ
8-  Năm TÂN TỴ
```

▪ Bạn suy nghĩ có đúng với kết quả như trên hay không? Bây giờ chúng tôi sẽ giải thích để các bạn hiểu rõ nhé.

▪ Thông thường các câu lệnh xuất trong một chương trình, trong một hàm, trong một lớp... sẽ xuất ra những câu tuần tự từ trên xuống. Nhưng do đặc thù của phương thức thiết lập và phương thức huỷ bỏ là cấp phát và lấy lại kí ức, nên những gì thuộc phương thức thiết lập thì thực hiện trước, do đó bạn thấy kết quả sẽ cho ra màn hình theo thứ tự như trên.

▪ Như các bạn biết việc cấp phát bộ nhớ sẽ đưa vào STACK, vậy cái nào cấp phát đầu tiên sẽ được bỏ vào STACK, kế đến là thứ hai được bỏ vào xếp chồng lên trên, và cứ thế sẽ tiếp tục. Do đó khi lấy lại, nó sẽ lấy cái trên cùng ra trước rồi đến cái trên cùng kế tiếp và như thế sẽ lấy lại cho đến cái cuối cùng, tức là cái cấp phát ban đầu. Do đó bạn nhìn trong chương trình có bốn phương thức thiết lập, mỗi phương thức có một câu chúc mừng, do đó sau khi bốn câu chúc mừng xuất hiện theo trình tự xong,

bạn sẽ thấy chương trình lấy cái sau cùng trước của phương thức huỷ bỏ cuối cùng (lúc này nó sẽ nằm trên cùng của STACK) . Nó sẽ lấy theo trình tự như sau:

TÂN TỴ - NHÂM NGỌ - QUÝ MUI
Năm QUÝ MÙI
Năm NHÂM NGỌ
Năm TÂN TỴ

▪ Các phương thức này với mục đích chính là cho bạn dùng vào việc cấp phát và thu hồi bộ nhớ, để bạn có thể sử dụng trong chương trình, nó sẽ mang lại hiệu quả là cách phân phối cũng như lấy lại khi cần lấy tránh dư thừa, chiếm chỗ làm chương trình của bạn có thể chạy chậm lại.

▪ Qua chương trình trên, chúng tôi giới thiệu đến các bạn để thấy trình tự thực hiện của phương thức để các bạn yên tâm sử dụng hai phương thức này trong các chương trình của bạn.

▪ Bây giờ thêm tập tin tiêu đề có tên <conio.h> để sử dụng hàm clrscr() và hàm getch().

+ Hàm clrscr() dùng để xoá màn hình.

+ Hàm getch() dùng để dừng chương trình giúp bạn có thể thấy kết quả, nếu không dùng hàm này thì muốn thấy kết quả sau khi chạy chương trình, bạn phải nhấn ALT+F5. Bạn sẽ có chương trình như sau:

```
#include<iostream.h>
#include<conio.h>
class chucmung1
 {
   public :
    chucmung1()
        {
          cout<<"Chuc mung ban mot nam moi an khang \n";
        }
    ~chucmung1()
        {
          cout<<"Nam TAN TY \n";
        }
 };
 class chucmung2 : public chucmung1
  {
    public :
```

```
        chucmung2()
         {
           cout<<"Chuc mung ban mot nam moi truong tho \n";
         }
       ~chucmung2()
         {
           cout<<"Nam NHAM NGO \n";
         }
};
class chucmung3 : public chucmung2
  {
    public :
        chucmung3()
         {
           cout<<"Chuc mung ban mot nam moi Hanh phuc \n";
         }
       ~chucmung3()
         {
           cout<< "Nam QUY MUI \n";
         }
};
class chucmung4 : public chucmung3
  {
    public :
        chucmung4()
         {
           cout<<"AN KHANG TRUONG THO VA HANH PHUC \n";
         }
       ~chucmung4()
         {
           cout<< " TAN TY - NHAM NGO - QUY MUI \n";
         }
};
void main()
  {
        clrscr();
       chucmung4 object;
        getch();
```

```
}
```

Những dòng đậm là mới thêm vào. Bây giờ chạy thử chương trình, bạn sẽ thấy như thế nào.

Nếu chạy chương trình, bạn sẽ thấy kết quả như sau :

```
Chúc mừng bạn một năm mới an khang
Chúc mừng bạn một năm mới trường thọ
Chúc mừng bạn một năm mới hạnh phúc
AN KHANG - TRƯỜNG THỌ - HẠNH PHÚC
```

- Bạn thử tìm hiểu xem sao!
- Trên là bạn tạo nhiều lớp dẫn xuất, lớp này sẽ thừa kế lần lượt các lớp vừa có trước đó.
- Bạn cũng có thể dùng một lớp dẫn xuất để thừa kế các lớp cơ sở có trước. Bạn có thể sửa lại chương trình như sau, kết quả vẫn không có gì thay đổi.

```
#include<iostream.h>
class chucmung1
 {
   public :
     chucmung1()
        {
          cout<<"Chuc mung ban mot nam moi an khang \n";
        }
     ~chucmung1()
        {
          cout<<"Nam TAN TY \n";
        }
 };
 class chucmung2
  {
     public :
          chucmung2()
           {
              cout<<"Chuc mung ban mot nam moi truong tho \n";
           }
         ~chucmung2()
```

```
        {
          cout<<"Nam NHAM NGO \n";
        }
};
class chucmung3
  {
    public :
        chucmung3()
        {
          cout<<"Chuc mung ban mot nam moi Hanh phuc \n";
        }
       ~chucmung3()
        {
          cout<<"Nam QUY MUI \n";
        }
};
class chucmung4 : public chucmung1, public chucmung2,
                          public chucmung3
  {
    public :
        chucmung4()
        {
          cout<<"AN KHANG - TRUONG THO - HANH PHUC \n";
        }
       ~chucmung4()
        {
          cout<< " TAN TY - NHAM NGO - QUY MUI \n";
        }
};
void main()
  {
      chucmung4 object;
  }
```

• Trong chương trình này, đầu tiên bạn xây dựng ba lớp cơ sở chucmung1, chucmung2 và chucmung3. Trong ba lớp này, các bạn cũng xây dựng trong mỗi lớp hai phương thức thiết lập và hủy bỏ. Cuối cùng bạn tạo một lớp dẫn xuất thứ tư có tên chucmung4, lớp này sẽ thừa kế ba lớp cơ sở đã có ở trên, bạn thực hiện bằng câu lệnh sau:

Chucmung4 : public chucmung1, public chucmung2, public chucmung3

- Lúc này chắc các bạn đã quen với việc tạo các lớp dẫn xuất thừa kế đơn, thừa kế bội rồi phải không.
- Sau đây chúng tôi giới thiệu đến các bạn thêm một ví dụ nữa để các bạn có thể hiểu rõ hơn về phương thức thiết lập và phương thức huỷ bỏ.

```
#include<iostream.h>
#include<conio.h>
class toan
 {
   private :
     int a;
     int b;
   public :
    toan(int , int);
   ~toan();
   int  cong ();
   int tru();
   int nhan();
   int chia();
 };
 toan::toan(int x, int y )
    {
        a=x;
        b=y;
    };
toan :: ~toan()
    {
        cout<<"\n";
        cout<<"BAN XEM BON KET QUA TREN  CO DUNG KHONG?";
    };
int toan :: cong()
   {
        return a+b;
   }
int toan :: tru()
   {
        return a-b;
   }
```

```
int toan :: nhan()
  {
      return a*b;
  }
int toan :: chia()
  {
      return a/b;
  }
main()
 {
    clrscr();
    toan ketqua(12, 18);
    cout<< "a+b= "<<ketqua.cong();
    cout<< "\n a-b= "<<ketqua.tru();
    cout<< "\n a*b= "<<ketqua.nhan();
    cout<< "\n a/b= "<<ketqua.chia();
   // getch();
    return 0;
 }
```

- Trong chương trình này, chúng tôi xây dựng một lớp Toan, trong lớp này, đầu tiên chúng tôi khai báo hai biến riêng của lớp có kiểu nguyên là a và b.
- Trong lớp này, chúng tôi xây dựng phương thức thiết lập và phương thức huỷ bỏ và các phương thức thành phần là:

Cong()
Tru()
Nhan()
Chia()

- Trong phương thức thiết lập, chúng tôi gán hai giá trị a=x và b=y.
- Trong phương thức huỷ bỏ, chúng tôi cho xuất hiện ra trên màn hình câu "BAN XEM BON KET QUA TREN CO DUNG KHONG?". Và chắc chắn câu này sẽ xuất hiện sau cùng, sau các thực hiện của các phương thức thiết lập và các phương thức thành phần. Bạn đã biết lý do tại sao rồi phải không.
- Kế tiếp là các phương thức thành phần cong, tru, nhan và chia.
- Trong hàm main(), khai báo đối tượng ketqua có kiểu toan, và chọn x=12, y=18

• Gọi tính toán các hàm cong , tru, nhan và chia với hai số 12 và 18.

• Chạy chương trình trên, bạn sẽ có kết quả như sau:

```
a+b=30
a-b=-6
a*b=226
a/b=0
```

BAN XEM BON PHEP TOAN TREN CO DUNG HAY KHONG?

Bạn thấy câu BAN XEM BON PHEP TOAN TREN CO DUNG HAY KHONG? sẽ xuất hiện sau cùng.

• Nếu trong chương trình, bạn dùng hàm getch(), bạn sẽ thấy kết quả sẽ không còn dòng ghi chú này nữa.

• Bây giờ bạn thay đổi phương thức huỷ bỏ lại như trong chương trình sau

```
#include<iostream.h>
#include<conio.h>
class toan
 {
   private :
     int a;
     int b;
   public :
    toan(int , int);
   ~toan();
   int  cong ();
   int tru();
   int nhan();
   int chia();
 };
 toan::toan(int x, int y)
     {
         a=x;
         b=y;
     };
 toan :: ~toan()
     {
                 ;
     };
```

```
int toan :: cong()
  {
      return a+b;
  }
int toan :: tru()
  {
      return a-b;
  }
int toan :: nhan()
  {
      return a*b;
  }
int toan :: chia()
  {
      return a/b;
  }
main()
 {
    clrscr();
    toan ketqua(12, 18);
    cout<< "a+b= "<<ketqua.cong();
    cout<< "\n a-b= "<<ketqua.tru();
    cout<< "\n a*b= "<<ketqua.nhan();
    cout<< "\n a/b= "<<ketqua.chia();
   // getch();
    return 0;
 }
```

▪ Nhìn vào hàm huỷ bỏ ta thấy không có tham số và không trả về giá trị nào, nó sẽ xoá bỏ đối tượng ketqua khi thấy đối tượng này không cần thiết nữa.

▪ Bây giờ chúng tôi gửi tới các bạn khái niệm về chồng hàm, chồng hàm trong C++ có nghĩa là ta có thể dùng một tên hàm cho nhiều công dụng khác nhau.

▪ Bạn có thể xem ví dụ sau để thấy được khái niệm này

```
#include<iostream.h>
#include<conio.h>
int cong(int, int);
```

```
float cong(float, float);
int trug(int, int);
float tru(float, float);
main()
 {
   clrscr();
   int a;
   int b;
   float c;
   float d;
   a=10;
   b=25;
   c=20.5;
   d=29.5;
   cout<<cong(a, b);
   cout<<cong(c, d);
   cout<<tru(a, b);
   cout<<tru(c, d);
   getch();
   return 0;
 }
 int cong (int a, int b)
   {
        return(a+b);
   }
 float cong(float c, float d)
   {
        cout<<"\n";
        return(c+d);
   }
int tru (int a, int b)
   {
        return(a-b);
   }
float tru(float c, float d)
   {
        cout<<"\n ";
        return(c-d);
```

}

- Trong chương trình trên ta có hai hàm **cong** và hai hàm **tru**
  + **Hàm int cong(int, int) có hai đối số là số nguyên và trả về một số nguyên.**
  + **Hàm float cong(float, float) có hai đối số là số thực và trả về một số thực.**
  + **Hàm int tru(int, int) có hai đối số là số nguyên và trả về một số nguyên.**
  + **Hàm float tru(float, float) có hai đối số là số thực và trả về một số thực.**
- Khi chạy thử chương trình trên, bạn sẽ có kết quả như sau:

35
50
-15
-9

- Bây giờ chúng tôi xin giới thiệu đến các bạn tính đa hình (polymorphism) trong C++.
- Tên gọi đa hình cũng giúp các bạn hình dung ra sự đa dạng và phong phú của nó. Trong lập trình hướng đối tượng của C++, bạn hiểu một cách nôm na là nó cho phép chúng ta gởi cùng một thông điệp nào đó đến nhiều đối tượng khác nhau cùng một lúc, nói một cách khác, thông điệp được gửi đi đến bất kỳ thực thể nào ở các lớp khác nhau.

*Ví dụ:*

- Bạn cần gửi một thông điệp tính toán chẳng hạn là phép toán cộng đến các đối tượng là

  Phép cộng hai số
  Phép cộng ba số
  Phép cộng bốn số .....

- Trong các đối tượng này đều có chung phương thức cộng, khi nhận được thông điệp, phép tính được tính toán và cho ra kết quả.
- Trong lập trình hướng đối tượng, tính đa hình được thể hiện qua khả năng cho phép mô tả những phương thức có tên giống nhau trong các lớp khác nhau. Để hình dung được phần nào những điều bạn sắp thấy, chúng tôi đã giới thiệu đến các bạn sự chồng hàm qua ví dụ ở trên.

▪ Chính đặc điểm này, giúp cho người lập trình không phải viết những cấu trúc rườm rà trong chương trình của mình, đồng thời nhờ tính đa hình này, các bạn thấy chương trình sáng sủa hơn và hay hơn.

▪ Tính đa hình của lập trình hướng đối tượng có được chính là nhờ vào các khái niệm như phương thức ảo (hàm ảo) và kỹ thuật kết nối động như những điều bạn sẽ thấy sau đây

**Phương thức ảo (virtual method)**

▪ Trong các phần trên của chương này, các bạn đã biết phần nào về lập trình hướng đối tượng, lúc này, các bạn cần phải biết cách sử dụng phương thức ảo, sự kết nối động để thấy được công dụng của nó là đối tượng sẽ nhận được chỉ thị cần thiết ngay khi chạy chương trình. Kỹ thuật nầy được gọi là sự kết nối động, nó cho phép chúng ta can thiệp vào sự hoạt động của những thực thể mà không cần biên dịch lại chương trình, chúng ta có thể truyền và nhận thông tin từ những đối tượng mới nầy giống như các đối tượng đã có.

▪ Để có được những điều hay và thuận lợi trên, trong khi xây dựng các lớp cơ sở, chúng ta sẽ cài phương thức ảo (virtual) với một tên nào đó có từ khoá virtual ở trước. Các bạn sẽ thấy trong các ví dụ bên dưới, trong các lớp dẫn xuất các bạn cũng khai báo các phương thức có tên giống như tên của phương thức ảo nhưng không có từ khoá virtual, khi thực hiện phương thức nào thông qua đối tượng tương ứng, kết quả của phương thức này sẽ đè lên phương thức ảo của lớp cơ sở.

▪ Trong lập trình hướng đối tượng thường gọi là thành phần ảo, đây sẽ là các phương thức sẽ mang nội dung cụ thể vào lúc chạy chương trình. Khi trình biên dịch biên dịch chương trình nguồn ra ngôn ngữ máy, các đoạn chương trình của phương thức ảo sẽ được đưa vào trong bảng các phương thức ảo. Vào thời điểm chương trình thực hiện, các phương thức ảo sẽ được lấy ra từ bảng phương thức ảo này. Chính vì vậy, kỹ thuật kết nối động đôi khi còn được gọi là kỹ thuật kết nối trễ, hay kết nối vào lúc chạy.

▪ Tính đa hình cho chúng ta thấy, một phương thức ảo không chỉ dành riêng hoặc chỉ ưu tiên cho một hai đối tượng nào mà nó có thể tác động đến nhiều đối tượng khác nhau tuỳ thuộc vào nội dung cụ thể của đối tượng đó.

▪ Trước khi viết một ví dụ cụ thể, chúng tôi có thể minh hoạ cho các bạn thấy qua mô hình cụ thể như sau:

▪ Thao tác làm đồng các xe khác nhau, vì vậy trong lớp cơ sở XE có một hành động chung đó là làm đồng. Tuy nhiên tùy theo mỗi loại xe mà chúng ta có cách làm đồng riêng, đòi hỏi phải có một kỹ thuật khác nhau, ví dụ đối với xe đạp sẽ khác với xe honđa sẽ khác với xe hơi, sẽ khác với xe lửa ... Bạn có thể nhìn qua sơ đồ sau:

XE
Làm đồng

| Xe đạp | Xe Honđa | Xe hơi | Xe lửa |
|---|---|---|---|
| Làm đồng | Làm đồng | Làm đồng | Làm đồng |

▪ Bạn thấy trong lớp cơ sở XE, chúng ta sẽ khai báo một phương thức ảo có tên là Làm đồng.

▪ Theo quan điểm của lập trình hướng đối tượng, hành động làm đồng cũng sẽ xuất hiện trong các lớp dẫn xuất Xe đạp, Xe Honđa, Xe hơi và Xe lửa.

▪ Với đối tượng của các lớp chúng ta có thể gọi trực tiếp hành động làm đồng để thực hiện.

*Ví dụ:* Phuongphap là đối tượng của lớp Xe đạp. Bạn có thể gọi trực tiếp hành động làm đồng như sau:

```
Phuongphap.Làm đồng();
```

▪ Dĩ nhiên là khi viết chương trình, các hành động và đối tượng bạn sẽ viết liền không dùng dấu và có khoảng trắng.

▪ Bạn sẽ thấy rõ hơn qua ví dụ sau, trong ví dụ này, chúng tôi gửi đến các bạn một phương thức ảo Cong(), phương thức ảo này sẽ được xác lập trong lớp cơ sở có tên là Tinhtoan

*Ghi chú:*

▪ Phương thức ảo được xác lập với cú pháp như sau:

```
virtual Tênhànhđộng()
```

▪ Các lớp dẫn xuất phepconghaiso, phepcongbaso và phepcongbonso được thừa kế từ lớp cơ sở có tên Tinhtoan.

Đối tượng ketquaconghaiso có kiểu phepconghaiso

Đối tượng ketquacongbaso có kiểu phepcongbaso

Đối tượng ketquacongbonso có kiểu phepcongbonso

- Trong ba lớp dẫn xuất trên, chúng tôi có xác lập phương thức cong() (có tên như phương thức ảo ở trong lớp dẫn xuất), chúng tôi sẽ gửi thông điệp cong() đến các đối tượng trong ba lớp khác nhau ở trên để tính toán. Chương trình được minh hoạ như sau:

```
#include<iostream.h>
#include<conio.h>
 class Tinhtoan
  {
      public :
      virtual void cong(void);
  };
 class phepconghaiso:public Tinhtoan
     {
          public :
            void cong(void);
     }ketquaconghaiso;
  void phepconghaiso:: cong(void)
     {
          int a;
          int b;
          cout<<"\n Moi ban nhap a= ";
          cin>>a;
          cout<<"\n Moi ban nhap b= ";
          cin>>b;
          int x;
          x=a+b;
          cout<<"a+b= "<<x;
     }
class phepcongbaso : public Tinhtoan
     {
          public :
            void cong(void);
     }ketquacongbaso
  void phepcongbaso :: cong(void)
    {
          int a;
```

```
        int b;
        int c;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
        int x;
        x=a+b+c;
        cout<<"a+b+c= "<<x;
    }
  class phepcongbonso : public Tinhtoan
    {
        public :
          void cong(void);
    }ketquacongbonso;
  void phepcongbonso :: cong(void)
    {
        int a;
        int b;
        int c;
        int d;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
        cout<<"Moi ban nhap d= ";
        cin>>d;
        int x;
        x=a+b+c+d;
        cout<<"a+b+c+d= "<<x;
    }
main()
  {
    int banchon;
```

```
do
 {
  cout<<"\n Ban chon 1, 2, 3 hoac 4?"
        << "\n 1- Cong hai so"
        << "\n 2- Cong ba so"
        << "\n 3- Cong bon so"
        << "\n 4- Thoat khoi chuong trinh";
  cin>>banchon;
  switch(banchon)
      {
        case 1:
          {
              ketquaconghaiso.cong();
                break;
          }
        case 2 :
          {
              ketquacongbaso.cong();
              break;
          }
        case 3 :
          {
              ketquacongbonso.cong();
              break;
          }
        case 4 : break;
        default : cout<<"Sai roi, chon 1, 2, 3 hoac 4. ";
      }
 } while(banchon!=4);
 return 0;
}
```

- Trong chương trình trên, đầu tiên bạn cài đặt một phương thức ảo có tên là cong() trong lớp cơ sở tinhtoan.
- Sau đó bạn cài đặt ba lớp dẫn xuất thừa kế lớp cơ sở tính toán có tên lần lượt là

Phepconghaiso
Phepcongbaso
Phepcongbonso

▪ Trong các lớp dẫn xuất này, mỗi lớp có một phương thức cong().

▪ Mỗi phương thức chỉ thuộc vào một lớp dẫn xuất tương ứng với code của chương trình khác nhau.

▪ Trong hàm main() là cách thể hiện thông điệp cong đến từng đối tượng và cho chúng ta kết quả theo như các số bạn nhập vào.

▪ Chạy thử chương trình trên, bạn sẽ thấy kết quả là đầu tiên sẽ xuất hiện một Menu

```
Ban chon 1, 2, 3 hoac 4?
    1- Cong hai so
    2- Cong ba so
    3- Cong bon so
    4- Thoat khoi chuong trinh
```

▪ Nếu bạn gõ số 1 và click Enter, bạn sẽ thấy trên màn hình cho bạn nhập lần lượt hai số a và b, sau khi nhập xong, chương trình sẽ cho kết quả và hiện lại menu trên.

▪ Tương tự nếu bạn nhập số 2, rồi gõ Enter, bạn sẽ thấy chương trình cho nhập lần lượt vào 3 số a, b và c, sau đó in ra kết quả và menu chương trình hiện ra.

▪ Tương tự như vậy với số 3, chương trình sẽ cho bạn nhập bốn số ...

▪ Nếu muốn thoát khỏi chương trình, bạn có thể nhập vào số 4, và gõ Enter. Bạn chạy thử và sẽ thấy.

▪ Bây giờ chúng ta thêm ba phương thức tru(), nhan() và chia() vào trong lớp phepcongbonso, và khai báo các phương thức này, chương trình như sau:

```
#include<iostream.h>
#include<conio.h>
class Tinhtoan
 {
     public :
          virtual void cong(void);
 };
class phepconghaiso:public Tinhtoan
    {
         public :
          void cong(void);
    }ketquaconghaiso;
void phepconghaiso:: cong(void)
```

```
    {
        int a;
        int b;
        cout<<"\n Moi ban nhap a= ";
        cin>>a;
        cout<<"\n Moi ban nhap b= ";
        cin>>b;
        int x;
        x=a+b;
        cout<<"a+b= "<<x;
    }
class phepcongbaso : public Tinhtoan
    {
        public :
            void cong(void);
    }ketquacongbaso;

void phepcongbaso :: cong(void)
  {
        int a;
        int b;
        int c;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
        int x;
        x=a+b+c;
        cout<<"a+b+c= "<<x;
  }
class phepcongbonso : public Tinhtoan
  {
        public :
            void cong(void);
            void tru(void);
            void nhan(void);
```

```
        void chia(void);
    }ketquacongbonso;
void phepcongbonso :: cong(void)
  {
        int a;
        int b;
        int c;
        int d;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
        cout<<"Moi ban nhap d= ";
        cin>>d;
        int x;
        x=a+b+c+d;
        cout<<"a+b+c+d= "<<x;
  }
void phepcongbonso :: tru(void)
 {
   int a;
   int b;
   int x;
   cout<<"\n Ban nhap vao so bi tru a= ";
   cin>>a;
   cout<<"\n Ban nhap vao so tru b= ";
   cin>>b;
   x=a-b;
   cout<<"a-b= "<<x;
 }

void phepcongbonso :: nhan(void)
 {
   int a;
   int b;
```

```
    int x;
    cout<<"\n Ban nhap vao so a= ";
    cin >> a;
    cout<<"\n Ban nhap vao so b= ";
    cin>>b;
    x=a*b;
    cout<<"a*b= "<<x;
  }

void phepcongbonso :: chia(void)
  {
    int a;
    int b;
    int x;
    cout<<"\n Ban nhap vao so bi chia a= ";
    cin>>a;
    cout<<"\n Ban nhap vao so chia b= ";
    cin>>b;
    x=a/b;
    cout<<"a/b= "<<x;
  }
main()
  {
    int banchon;
    do
     {
      cout << "\n Ban chon 1, 2, 3, 4, 5, 6 hay 7?"
           << "\n 1- Cong hai so"
           << "\n 2- Cong ba so"
           << "\n 3- Cong bon so"
           << "\n 4- Tru"
           << "\n 5- Nhan"
           << "\n 6- Chia"
           << "\n 7- Thoat khoi chuong trinh";
      cin>>banchon;
      switch(banchon)
          {
```

```
            case 1:
              {
                    ketquaconghaiso.cong();
                      break;
              }
            case 2 :
              {
                    ketquacongbaso.cong();
                    break;
              }
            case 3 :
              {
                    ketquacongbonso.cong();
                    break;
              }
            case 4:
              {
                    ketquacongbonso.tru();
                    break;
              }
            case 5 :
              {
                    ketquacongbonso.nhan();
                     break;
              }
            case 6 :
              {
                    ketquacongbonso.chia();
                    break;
              }
            case 7 : break;
            default : cout<<"Sai roi, chon 1, 2, 3, 4, 5, 6 hoac 7.";
          }
  } while(banchon!=7);
 return 0;
}
```

• Qua chương trình trên, bạn thấy những chữ đậm là chúng tôi mới thêm vào, sửa lại một chút trong phát biểu Switch.

• Vậy là các bạn thấy chúng tôi thêm vào lớp phepcongbonso, thêm ba phương thức tru(), nhan() và chia(), ba phương thức này thực hiện độc lập, không có dạng đa hình, không dùng phương thức ảo. Chúng tôi muốn đưa ra cho các bạn có thể thấy thêm và có thể ứng dụng trong các chương trình của các bạn sau này.

• Nếu chạy thử chương trình, bạn sẽ thấy menu hiện ra như sau:

```
Ban chon 1, 2, 3, 4, 5, 6 hay 7?
        1- Cong hai so
        2- Cong ba so
        3- Cong bon so
        4- Tru
        5- Nhan
        6- Chia
        7- Thoat khoi chuong trinh
```

• Khi bạn gõ vào số tương ứng, chương trình sẽ cho bạn các thao tác nhập và cho ra kết quả. Bạn tự kiểm tra nhé!

• Bây giờ chúng tôi thử cài thêm một lớp mới và tạo một phương thức ảo mới trong lớp này, các bạn xem chương trình sau:

```
#include<iostream.h>
#include<conio.h>
class Tinhtoan
 {
     public :
         virtual void cong(void);
 };
class Tinhtoanhai
 {
     public :
         virtual void nhan(void);
 };
class phepconghaiso:public Tinhtoan
    {
        public :
          void cong(void);
    }ketquaconghaiso;
```

```
void phepconghaiso:: cong(void)
   {
        int a;
        int b;
        cout<<"\n Moi ban nhap a= ";
        cin>>a;
        cout<<"\n Moi ban nhap b= ";
        cin>>b;
        int x;
        x=a+b;
        cout<<"a+b= "<<x;
   }
class phepcongbaso : public Tinhtoan
   {
        public :
           void cong(void);
   }ketquacongbaso;
void phepcongbaso :: cong(void)
  {
        int a;
        int b;
        int c;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
        int x;
        x=a+b+c;
        cout<<"a+b+c= "<<x;
  }
class phepcongbonso : public Tinhtoan
  {
        public :
           void cong(void);
  }ketquacongbonso;
```

```
void phepcongbonso :: cong(void)
  {
        int a;
        int b;
        int c;
        int d;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
        cout<<"Moi ban nhap d= ";
        cin>>d;
        int x;
        x=a+b+c+d;
        cout<<"a+b+c+d= "<<x;
  }
class phepnhanhaiso:public Tinhtoanhai
  {
        public :
          void nhan(void);

  }ketquanhanhaiso;

void phepnhanhaiso:: nhan(void)
  {
        int a;
        int b;
        cout<<"\n Moi ban nhap a= ";
        cin>>a;
        cout<<"\n Moi ban nhap b= ";
        cin>>b;
        int x;
        x=a*b;
        cout<<"a*b= "<<x;
  }
```

```
class phepnhanbaso : public Tinhtoanhai
    {
        public :
            void nhan(void);
    }ketquanhanbaso;
 void phepnhanbaso :: nhan(void)
   {
        int a;
        int b;
        int c;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
        int x;
        x=a*b*c;
        cout<<"a*b*c= "<<x;
   }
class phepnhanbonso : public Tinhtoanhai
    {
        public :
            void nhan(void);
    }ketquanhanbonso;

 void phepnhanbonso :: nhan(void)
   {
        int a;
        int b;
        int c;
        int d;
        cout<<"Moi ban nhap a= ";
        cin>>a;
        cout<<"Moi ban nhap b= ";
        cin>>b;
        cout<<"Moi ban nhap c= ";
        cin>>c;
```

```
        cout<<"Moi ban nhap d= ";
        cin>>d;
        int x;
        x=a*b*c*d;
        cout<<"a*b*c*d= "<<x;
    }
main()
 {
   int banchon;
   do
    {
     cout<<"\n Ban chon 1, 2, 3, 4, 5, 6  hay 7?"
          << "\n 1- Cong hai so"
          << "\n 2- Cong ba so"
          << "\n 3- Cong bon so"
          << "\n 4- Nhan hai so"
          << "\n 5- Nhan ba so"
          << "\n 6- Nhan bon so"
          << "\n 7- Thoat khoi chuong trinh";
    cin>>banchon;
    switch(banchon)
        {
         case 1:
           {
                ketquaconghaiso.cong();
                  break;
           }
         case 2 :
           {
                ketquacongbaso.cong();
                break;
           }
         case 3 :
           {
                ketquacongbonso.cong();
                break;
           }
```

```
            case 4:
              {
                    ketquanhanhaiso.nhan();
                      break;
              }
            case 5 :
              {
                    ketquanhanbaso.nhan();
                    break;
              }
            case 6 :
              {
                    ketquanhanbonso.nhan();
                    break;
              }

            case 7 : break;
            default : cout<<"Sai roi, chon 1, 2, 3, 4, 5, 6 hoac 7.";
          }
    } while(banchon!=7);
  return 0;
}
```

▪ Trong chương trình này, chúng tôi tạo thêm một lớp cơ sở nữa, đó là lớp Tinhtoanhai. Trong lớp này chúng tôi cài một phương thức ảo có tên là nhân, chúng tôi cài với cú pháp như sau :

```
virtual  nhan(void);
```

▪ Chúng tôi tạo thêm ba lớp dẫn xuất có tên là

Phepnhanhaiso

Phepnhanbaso

Phepnhanbonso

▪ Ba lớp này thừa kế lớp Tinhtoanhai, trong mỗi lớp có cài một phương thức nhân tương ứng phù hợp với ứng dụng của nó, bạn có thể truyền thông điệp nhân đến cho các đối tượng như trong chương trình trước.

▪ Bạn thử cài đặt trong một lớp có hai phương thức ảo, bạn sẽ thấy trình biên dịch sẽ báo gì khi bạn biên dịch chương trình.

Bạn có thể tạo thêm một phương thức ảo có tên nhan() trong lớp Tinhtoan thay vì phải tạo thêm một lớp Tinhtoanhai như trên. Bạn thử để thấy chương trình báo như thế nào nhé!

# Bài 11 *(ngày 14)*

# Một số hàm trong C++

☞ **Một số hàm trong Toán học**

1- **abs**(num)

+ Hàm này cho kết quả là trị tuyệt đối của num

*Ví dụ:*

abs(-12)=12

abs(4)=4

2- **exp**(num)

+ Kết quả là e mũ num ($e^{num}$)

ví dụ: exp(2)=$e^2$

3- **log**(num)

+ Kết quả là Logarit tự nhiên của num.

4- **log10**(num)

+ Logarit của num (với cơ số 10)

5- **sqrt**(num)

+ Kết quả là căn bậc hai của num

Ví dụ: sqrt(4)=2

6- **pow**(num1, num2)

+ Kết quả là num1 lũy thừa num2

Ví dụ: pow(2, 3)=8

7- **cos**(num)

+ Kết quả là cos của (num)

8- **sin**(num)

+ Kết quả là sin của num

9- **asin**( num )

+ Kết quả là nghịch đảo của hàm sin

10- **acos**(num)

+ Kết quả nghịch đảo của hàm cosin

11- **tan**(num)

+ Tang của num

12- **atan**(num)

+ Nghịch đảo của hàm Tang

13- **ceil**(num)

+ Lấy phần nguyên của num cộng thêm 1

Ví dụ:

ceil(-12.45)=12

ceil(4.34)=5

14- **floor**(num)

+ Lấy phần nguyên của num

*Ví dụ:*

floor(-12.45)=-13

floor(4.45)=4

Trên đây là một số hàm hay dùng trong toán học, khi sử dụng những hàm này, chúng ta phải khai báo tập tin tiêu đề <math.h>.

Các hàm toán học ở trên, đối số có kiểu double và trị trả về cũng có kiểu double.

### ☞ Một số hàm hay dùng trong chuỗi

1- **strlen**(ch)

+ Cho biết chiều dài của chuỗi ch

*Ví dụ:*

ch="LaptrinhC++"

Strlen(ch)=10

2- **strcat**(hằngchuỗi1, hằngchuỗi2)

+ Nối hằng chuỗi 2 vào cuối hằng chuỗi 1

*Ví dụ:*

ch1="Laptrinh"

ch2="C++"

Strcat(ch1, ch2)=LaptrinhC++

3-**strcpy** (biến, hằngchuỗi)

+ Gán hằng chuỗi cho biến

*Ví dụ:*

strcpy(x, "LaptrinhC++")

Vậy x=LaptrinhC++

Bạn có thể tham khảo thêm về hàm bằng cách vào phần Help của C++, có những hàm người ta có cho ví dụ trong đó, nếu trong khi viết chương trình có dùng hàm mới, chúng tôi sẽ giải thích.

Phần dùng chương trình con bằng cách dùng hàm hoặc thủ tục, chúng tôi sẽ giới thiệu trong phần bài tập.

# CÁC CHƯƠNG TRÌNH MẪU ÁP DỤNG PHẦN LÝ THUYẾT

## *(Từ ngày 15 đến ngày 21)*

## ☎ Phần 1

## Bài tập đơn giản

### ✍ In ra trên màn hình một câu chào

Bài tập đơn giản đầu tiên này, chúng tôi sẽ viết một chương trình để xuất hiện ra màn hình một câu chào

LAP TRINH HUONG DOI TUONG C++
DANG CHAO DON CAC BAN

```
// Gioithieu.cpp
 #include <iostream.h>
 #include <conio.h>
 main()
 {
   cout<<"LAP TRINH HUONG DOI TUONG C++  \n";
   cout<<"DANG CHAO DON CAC BAN";
   getch();
   return(0);
 }
```

⇨ Bạn lưu chương trình bằng cách nhấn phím F2, trong hộp thoại hiện ra, chọn ổ đĩa và thư mục cần lưu, đặt tên cho chương trình và lưu. Bạn cũng có thể vào menu File, chọn Save và thực hiện như trên. Bạn nhớ lưu thường xuyên để tránh trường hợp mất điện có thể xảy ra.

⇨ Trong chương trình này không có dùng một biến nào cả.

⇨ Dấu // dùng để ghi chú tên của chương trình. Trình biên dịch sẽ bỏ qua dòng ghi chú theo sau dấu // này.

⇨ Trong ngôn ngữ C++ dùng kí tự là chữ thường, bạn không thể dùng các chữ in hoa. Ví dụ nếu bạn khai báo Include (*chữ i là chữ in thì không được*) riêng các chuỗi chứa trong dấu nháy kép thì được.

⇨ Khai báo tập tin tiêu đề iostream.h vì tập tin này chứa dòng xuất cout.

⇨ Khai báo tập tin tiêu đề conio.h vì tập tin này chứa hàm getch(). Hàm này sẽ cho bạn thấy kết quả hiện trên màn hình, nếu không có hàm này, muốn thấy được kết quả của chương trình, bạn phải nhấn ALT+F5.

⇨ Hàm main() bắt buộc phải có trong mọi chương trình C++. Hàm này mở đầu bằng dấu móc

```
{
  Và kết thúc bằng
  return (0);
}
```

⇨ Dòng xuất cout và toán tử << sẽ cho các thông tin trong hai dấu kép hiện ra trên màn hình.

⇨ '\n' dùng để xuống dòng.

⇨ Dòng lệnh return(0) cho hệ thống biết chương trình trả quyền điều khiển về cho hệ điều hành, và giá trị 0 như là một tín hiệu báo cho hệ điều hành biết chương trình kết thúc thành công.

⇨ Nếu không thay đổi, khi nhấn Ctrl+F9 để chạy thử chương trình, bạn sẽ thấy trên màn hình sẽ có dòng thông báo như sau:

```
LAP TRINH HUONG DOI TUONG C++
DANG CHAO DON CAC BAN
```

Ghi chú:

Trong C++, bạn đánh chữ thường, không dùng chữ in hoa.

# ☎ Phần 2

## Bài tập sử dụng IF ... Then hoặc IF ... Then ... Else

## ✍ Làm toán cộng trừ trên máy

```
// Lamtoan.cpp
#include <iostream.h>
#include <conio.h>
int a;
int b;
int c;
int x;
int y;
int z;
// Ban co the khai bao int  a,b,c,x,y,z;
main()
{
  clrscr();
  cout<<"Be nhap so a= ";
  cin>>a;
  cout<<"Be nhap so b= ";
  cin>>b;
  cout<<"Be nhap so c= ";
  cin>>c;
  // Nhap cac gia tri tinh nham
  cout<<"Do be a+b+c bang bao nhieu? x=  ";
  cin>>x;
  cout<<"Do be a+b-c bang bao nhieu? y=  ";
  cin>>y;
  cout<<"Do be a-b+c bang bao nhieu? z=  ";
  cin>>z;
// Truong hop a+b+c
```

```
  if (x==a+b+c)
    cout<<"Hoan ho Be da lam dung, vi a+b+c= "<<x<<'\n';
  else
    cout<<"Rat tiec Be da lam sai \n ";

  // Truong hop a+b-c
  if (y==a+b-c)
    cout<<"Hoan ho Be da lam dung, vi a+b-c= "<<y<<'\n';
  else
    cout<<"Rat tiec Be da lam sai \n ";
  // Truong hop a-b+c
  if (z==a-b+c)
    cout<<"Hoan ho Be da lam dung, vi a-b+c= "<<z<<'\n';
  else
    cout<<"Rat tiec Be da lam sai \n ";
  getch();
  return(0);
}
```

⇨ Trong chương trình trên, bạn thấy chúng tôi khai báo 6 biến có kiểu là số nguyên, đó là các biến a, b, c, x, y và z.

⇨ Hàm clrscr() sẽ xoá màn hình, thông tin sẽ hiện góc trái trên cùng của màn hình. Tập tin tiêu đề conio.h chứa hàm này.

⇨ Đầu tiên bạn nhập giá trị của biến a, b và c nhờ dòng nhập cin cùng với toán tử >>.

⇨ Sau đó là nhập các kết quả tính nhẩm của a+b+c, a+b–c và a–b+c nhờ dòng nhập cin và toán tử >>.

⇨ Các phát biểu if sẽ kiểm tra điều kiện nếu thoả thì thực hiện lời phát biểu theo sau (*trước else*), ngược lại sẽ thực hiện phát biểu theo sau else.

```
If  <điều kiện >
    ..............
Else
    ..............
```

⇨ Trong C++ kí hiệu = là toán tử gán, nếu biến bằng biểu thức nào đó, chúng ta phải dùng kí hiệu == (*hai dấu =*), do đó trong điều kiện của if ta phải dùng kí hiệu ==

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Bạn sẽ thấy trên góc trái phía trên của màn hình, thông tin hiện ra từng dòng, sau khi nhập xong một số, gõ Enter, bạn sẽ thấy xuất hiện dòng kế tiếp... giả sử bạn nhập vào các số như sau:

```
Be nhap so a:  2
Be nhap so b:  2
Be nhap so c:  2
Do be a+b+c bang bao nhieu?  x=6
Do be a+b-c bang bao nhieu?  y=2
Do be a-b+c bang bao nhieu?  z=2
```

⊙ Nếu nhập như vậy, bạn sẽ có thông báo trên màn hình như sau

```
Hoan ho be da lam dung, vi a+b+c=6
Hoan ho be da lam dung, vi a+b-c=2
Hoan ho be da lam dung, vi a-b+c=2
```

⊙ Kiểm tra lại bạn thấy:

```
a+b+c=2+2+2=6 Đúng
a+b-c=2+2-2=2 Đúng
a-b+c=2-2+2=2 Đúng
```

⊙ Vì ba điều kiện đều đúng, nên ba câu lệnh sau điều kiện của if trước else sẽ thực hiện.

# ✍ Kiểm tra tính hợp lệ của tam giác

Chương trình này, chúng tôi dựa vào định lý về mối liên hệ giữa 3 cạnh của một tam giác để thực hiện chương trình. Định lý như sau:

- *Định lý 1:*

Trong một tam giác, tổng độ dài hai cạnh bất kỳ bao giờ cũng lớn hơn độ dài của cạnh còn lại.

- *Định lý 2:*

Trong một tam giác, hiệu độ dài hai cạnh bất kỳ bao giờ cũng nhỏ hơn độ dài cạnh còn lại.

+ *Tổng quát:*

"Trong một tam giác, mỗi cạnh lớn hơn hiệu và nhỏ hơn tổng độ dài hai cạnh kia"

- Trong bài này, khi chúng ta nhập vào độ dài 3 cạnh bất kỳ của một tam giác trong 3 TextBox, sau đó click vào nút xem kết quả, chương trình sẽ báo là 3 cạnh bạn vừa nhập có thoả định lý 1 và định lý 2 hay không.
- Nếu không thoả một trong hai định lý, hoặc có một cạnh nhỏ hơn 0 thì sẽ báo là không thoả (*Không có tam giác nào có độ dài 3 cạnh như vậy*).
- Nếu thoả thì sẽ tính chu vi của tam giác là:

**a+b+c**

- Đồng thời tính diện tích của tam giác theo công thức sau:

**S=Căn bậc hai của p(p–a)(p– b )(p–c)** với **p=(a+b+c)/2**

```
// Tamgiac.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
float a;
float b;
float c;
float p;
main()
{
  cout<<"Moi ban nhap vao do dai canh a:";
  cin>>a;
  cout<<"Moi ban nhap vao do dai canh b:";
  cin>>b;
  cout<<"Moi ban nhap vao do dai canh c:";
  cin>>c;
  p=(a+b+c)/2;

   if ((a+b>c)&&(c+b>a)&&(a+c>b)&&(a>0)&&(b>0)&&(c>0))
       {
         cout<<"Ket qua kiem tra la thoa dieu kien "<< '\n';
         cout<<"Chu vi cua tam giac la: "<<a+b+c<<'\n';
         cout<<"Dien tich cua tam giac la:"<<sqrt(p*(p-a)*(p-b)*(p-c));
       }
       else
```

```
            cout<<"Ket qua kiem tra khong thoa.";
        getch();
        return 0;
}
```

⇨ Trong bài này các bạn dùng toán tử && (AND) và hàm abs() tính giá trị tuyệt đối.

⇨ Trong điều kiện của phát biểu If, bạn dùng toán tử && để nối các biểu thức điều kiện, tức là tất cả các điều kiện đều thoả thì mới thực hiện các lệnh theo sau điều kiện (trước else).

⇨ Bạn thấy sau ĐK có ba câu lệnh, khi cần thực hiện từ hai lệnh trở lên của một điều kiện nào đó, bạn phải dùng:

```
{
    ........
    ........
}
```

để bao các câu lệnh đó.

⇨ Nếu thoả điều kiện, thì thông báo chu vi và diện tích của tam giác qua công thức tính được viết bằng ngôn ngữ C++.

⇨ Hàm sqrt() sẽ tính căn bậc hai của đối số trong ngoặc.

*Ví dụ:* x=sqrt(4)=2

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Bạn nhập lần lượt cạnh a=5, b=10 và c=12, bạn sẽ có thông tin như sau:

```
Moi ban nhap vao do dai canh a: 5
Moi ban nhap vao do dai canh b: 10
Moi ban nhap vao do dai canh c: 12
```

⊙ Khi nhập xong, kết quả thông báo sẽ là:

```
Ket qua kiem tra la thoa dieu kien
Chu vi cua tam giac la: 27
Dien tich cua tam giac la: 389.19147
```

⊙ Vì điều kiện của If thoả nên ta có các thông báo như trên, các thông báo này là nhờ các dòng xuất cout và toán tử <<.

⊙ Bây giờ bạn thử chạy lại chương trình và nhập a=1, b=2 và c=3.3 Bạn sẽ có kết quả như sau:

Moi ban nhap vao do dai canh a: 1
Moi ban nhap vao do dai canh b: 2
Moi ban nhap vao do dai canh c: 3.3

⊙ Kết quả thông báo sẽ là: Ket qua kiem tra khong thoa.

## ✍ Giải phương trình bậc 2 đủ

```
// PhuongTrinhBac2.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
float a,b,c;
float Delta;
float x1,x2;
main()
{
    clrscr();
    cout<<"Ban nhap he so a (a khac 0):";
    cin>>a;
    cout<<"Ban nhap he so b (b khac khong):";
    cin>>b;
    cout<<"Ban nhap he so c (c khac 0):";
    cin>>c;
    Delta=(b*b)-(4*a*c);
    if (Delta<0)
        cout<<"Phuong trinh vo nghiem";
    else
        if (Delta==0)
            cout<<"Phuong trinh co nghiem kep x1=x2="<<-(b/(2*a));
        else
            cout << "Phuong trinh co hai nghiem x1="<<
                (-b-sqrt(Delta))/(2*a)<<"x2="<<(-b+sqrt(Delta))/(2*a);
    getch();
    return 0;
}
```

## ✍ Kết luận có phải là năm nhuận

```
// Namnhuan.cpp (giả sử năm nhuận là năm chia hết cho 4)
#include <iostream.h>
#include <conio.h>
int n;
main()
{
    clrscr();
    cout << "Ban nhap vao nam can kiem tra:";
    cin >> n;
    if (n%4==0)
        cout<<"Nam"<<n<<" la mot nam nhuan.";
    else
        cout<<"Nam"<<n<<"khong phai la mot nam nhuan.";
    getch();
    return 0;
}
```

⇨ Trong bài này bạn chỉ dùng các điều kiện như đầu bài đã cho, để kiểm tra xem một năm nhập vào có phải là một năm nhuận hay không.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình, nhập vào lần lượt một số năm mà bạn tự chọn, bạn thấy chương trình báo ra rất là chính xác.

## ✍ Kiểm tra có phải là năm tổ chức Mundial hay không

☞ Cúp bóng đá thế giới Mundial được tổ chức lần đầu tiên vào năm 1930 và sau đó cứ 4 năm tổ chức một lần. Hãy viết chương trình cho nhập vào 1 năm và cho biết năm đó có tổ chức Mundial hay không, và đây là lần tổ chức thứ mấy, nếu nhập năm nhỏ hơn 1930 chương trình phải thông báo "Năm... không tổ chức giải Mundial"

*Ghi chú:* ở đây chúng ta xem như năm 1942 và 1946 cũng có tổ chức nhưng không thực hiện được. Sau khi thực hiện xong chương trình này, các bạn thử sửa lại cho đúng là năm 1942 và 1946 không có tổ chức.

**// Muldial.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,n,m;
main()
{
    clrscr();
    cout<<"Ban nhap nam can kiem tra (năm phải hợp lệ):";
    cin>>n;
    if (((n-1930)>=0)&&((n-1930)%4==0))
        {
            i=n-1930;
            m=i/4;
            cout<<"Nam"<<n<<"co to chuc Muldial va day la lan to
                        chuc thu:"<<m+1;
        }
    else
            cout<<"Nam"<<n<<"Khong co to chuc Muldial";
    getch();
    return 0;
}
```

◎ Chạy thử chương trình, bạn nhập n=2000, bạn thấy thông báo, năm 2000 không có tổ chức Muldial. Nếu bạn nhập n=2002, bạn thấy chương trình thông báo năm 2002 có tổ chức Muldial, và đây là lần tổ chức thứ 19.

# Tính cấp số cộng, cấp số nhân

⇨ Chương trình này sẽ cho người sử dụng nhập vào ba số phân biệt a, b và c theo một thứ tự. Kiểm tra xem có phải là một cấp số cộng hay không.

⇨ Nếu là một cấp số cộng, thì cấp số cộng này là cấp số cộng tiến hay là một cấp số cộng thoái.

⇨ Tương tự như trên, nhưng là cấp số nhân.

*Định nghĩa:* Cấp số cộng là một dãy những con số được xếp theo một thứ tự nhất định, trong đó mỗi số hạng bằng số hạng đứng trước cộng với một hằng số khác không.

$$U_n=U_{n-1}+r$$

Hằng số r (r khác 0) gọi là công sai của cấp số cộng.

⇨ Nếu r>0: Trị số các số hạng lớn dần, ta có cấp số cộng tiến.

⇨ Nếu r<0: Trị số các số hạng nhỏ dần, ta có cấp số cộng thoái.

*Ví dụ:*

1 3 5 7 9 ..... r = 2 > 0 → Cấp số cộng tiến

10 7 4 1 -2 ..... r = -3 < 0 → Cấp số cộng thoái

*Định lý:* Ba số phân biệt a, b, c theo thứ tự họp thành một cấp số cộng nếu và chỉ nếu:

**2b=a+c** (b là trung bình cộng của a và c)

*Định nghĩa:* Cấp số nhân là một dãy những con số được xếp theo một thứ tự nhất định, trong đó mỗi số hạng bằng số hạng đứng trước nhân với một hằng số khác 0 và 1.

$$U_n=U_{n-1}.q$$

Hằng số q (q<>0 và q<>1) gọi là công bội của cấp số nhân.

⇨ Nếu $u_1>0$ và q>1: Trị số của các số hạng lớn dần, ta có một cấp số nhân tiến.

⇨ Nếu $u_1>0$ và 0<q<1: Trị số của các số hạnh nhỏ dần, ta có một cấp số nhân thoái.

⇨ Nếu q<0: Những số hạng của cấp số lần lượt thay đổi dấu.

*Ví dụ:*

2/ 3 2 6 18 54 ... q = 3 → Một cấp số nhân tiến

4 2 1/ 2 1/4 1/8 ... q =1/2 → 0<q<1 cấp số nhân thoái

7 -7 7 -7 7 .... q = - 1

*Định lý:* Ba số phân biệt, không triệt tiêu a, b , c theo thứ tự họp thành một cấp số nhân nếu và chỉ nếu:

**$b^2$=a.c** (b là trung bình nhân của a và c)

```
// CapSoCong.cpp
#include <iostream.h>
#include <conio.h>
float a,b,c;
main()
{
   // Nhap vao ba so a, b va c
```

```
    cout<<"Ban nhap vao so a";
    cin>>a;
    cout<<"Ban nhap vao so b";
    cin>>b;
    cout<<"Ban nhap vao so c";
    cin>>c;
    // Kiem tra xem co phai la cap so cong hay khong
    if ((a+c==2*b)&&(b-a!=0))
     {
        if (b-a>0)
          {
            cout<<"Ba so"<<a<<"  "<<b<<"  "<<c<<"
                    theo thu tu nhu vay la mot cap so cong";
            cout<<"Day la mot cap so cong tien, voi cong sai la:"<<b-a;
          }
        if (b-a<0)
          {
            cout<<"Ba so"<<a<<"  "<<b<<"  "<<c<<
                    " theo thu tu nhu vay la mot cap so cong";
            cout<<"Day la mot cap so cong thoai, voi cong sai la:"<<b-a;
          }
     }
   else
     cout<<"Ba so"<<a<<"  "<<b<<"  "<<c<<
            "theo thu tu nhu vay khong phai la mot cap so cong";
 getch();
 return 0;
}
```

⇨ Khai báo các số a, b và c là số thực

⇨ Nhập vào 3 số a, b và c theo thứ tự. Sau đó kiểm tra xem ba số nhập vào theo thứ tự a, b , c có phải là một cấp số cộng hay không.

⇨ Trong bài này chúng tôi dựa vào định nghĩa và định lý về cấp số cộng như đã cho ở trên. Để viết chương trình, các điều kiện trong các phát biểu của If chính là các phát biểu của định nghĩa và định lý, các bạn tự tìm hiểu nhé.

```
// CapSoNhan.cpp
#include <iostream.h>
#include <conio.h>
float a,b,c;
main()
{
   // Nhap vao ba so a, b va c theo thu tu
   clrscr();
   cout << "Ban nhap vao so a";
   cin >> a;
   cout << "Ban nhap vao so b";
   cin >> b;
   cout << "Ban nhap vao so c";
   cin >> c;
  // Kiem tra xem co phai la cap so nhan hay khong
 if ((a==0)&&(b==0)&&(c==0))
   cout << "Day khong phai la mot cap so nhan";
else
   {
       if ((a*c==b*b)&&((b/a)!=0)&&(b/a!=1))
          {
              if ((b/a)>1)
                 {
                    cout<<"Ba so"<<a<<" "<<b<<" "<<c <<"theo thu
                              tu nhu vay la mot cap so nhan";
                    cout<<"Day la mot cap so nhan tien, voi cong boi
                              la:"<<b/a;
                 }
           if ((b/a<1)&&(b/a>0))
                {
                   cout<<"Ba so"<<a<<" "<<b<<" "<<c<<"theo
                              thu tu nhu vay la mot cap so nhan"<<"\n";
                   cout<<"Day la mot cap so nhan thoai, voi cong boi
                              la:"<<b/a;
                }
           if ((b/a)<0)
```

```
                {
                    cout<<"Ba so"<<a<<" "<<b<<" "<<c<<" theo
                            thu tu nhu vay la mot cap so nhan"<<"\n";
                    cout << "Day la mot cap so nhan, voi nhung so lan
                            luot thay doi dau, voi cong boi la:"<<b/a;
                }
            }
        else
            cout<<"Ba so "<<a<<" "<<b<<"  "<<c<<"theo thu tu
                    nhu vay khong phai la mot cap so nhan";
    }
    getch();
    return 0;
}
```

⇨ Chạy thử chương trình Cấp số cộng

⊙ Nhập vào các số a, b và c là: a=2, b=4 và c=6. Bạn sẽ có kết quả như sau:

```
Ba so  2, 4, 6
theo thu tu nhu vay la mot cap so cong tien, cong sai la 2.
```

⊙ Chạy lại chương trình và nhập a=16, b=14 và c=12. Bạn sẽ có kết quả

```
Ba so  16, 14, 12
theo thu tu nhu vay la mot cap so cong thoai, cong sai la -2
```

⊙ Chạy lại chương trình và nhập a=2, b=4 và c=12. Bạn sẽ có kết quả như sau:

```
Ba so  2, 4, 12
theo thu tu nhu vay khong phai la mot cap so cong
```

⇨ Chạy thử chương trình cấp số nhân:

⊙ Chạy thử chương trình cấp số nhân và nhập a=1, b=5 và c=25. Bạn sẽ có kết quả như sau:

```
Ba so  1, 5, 25
theo thu tu nhu vay la mot cap so nhan tien, cong boi la: 5
```

⊙ Chạy lại chương trình cấp số nhân và nhập a=0.5, b=2 và c=8. Bạn sẽ có kết quả như sau:

Ba so 0.5, 2, 8

theo thu tu nhu vay la mot cap so nhan tien, cong boi la: 4

⊙ Chạy thử chương trình cấp số nhân và nhập a=12, b=6 và c=3 Bạn sẽ có kết quả như sau:

Ba so 12, 6, 3

theo thu tu nhu vay la mot cap so nhan thoai,cong boi la: 0.5

⊙ Thử chương trình cấp số nhân và nhập a=1, b=-1 và c=1 Bạn sẽ có kết quả như sau:

Ba so 1, -1, 1

theo thu tu nhu vay la mot cap so nhan thoai,cong boi la: -1

# Tính Logarit

- **Của số a với cơ số là số c.**
- **Của số b với cơ số là số c.**
- **Của a nhân b với cơ số là c.**
- **Của a chia b với cơ số là c.**

Trong bài này, chúng tôi giới thiệu đến các bạn chương trình tính Logarit của một số, với số và cơ số được nhập vào từ người sử dụng.

Để tính Logarit của một tích, của một thương, người sử dụng sẽ nhập vào cơ số bất kỳ thoả điều kiện về cơ số, đó là cơ số phải lớn hơn 0 phải khác 1. Nhập số a, số b tùy ý, số này phải lớn hơn 0.

*Hướng dẫn*

$Log_c a$

thì c gọi là cơ số. c>0 và c<>1

a gọi là số a. ĐK a>0

*Ví dụ:* Cho $Log_2 8=x$

Trong trường hợp này Cơ số bằng 2

Số a=8

Vậy $2^x=8$

suy ra x=3

$Log_c(a)=Log(a)/Log(c)$

$Log_c(a*b)=Log_c a+Log_c b$

$Log_c(a/b)=Log_c a-Log_c b$

☞ **Chương trình a cơ số c**

```
// Tinh_Logarit_ a_coso_c.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int a,c;
float z;
main()
{
    clrscr();
    cout<<"Ban nhap so a= ";
    cin>>a;
    cout<<"Ban nhap vao co so c= ";
    cin>>c;
     if ((a>0)&&(c>0)&&(c!=1))
        {
           z=log(a)/log(c);
           cout<<"Logarit co so "<<c<<" cua "<<a<<" = "<<z;
        }
    else
          cout<<"Khong thoa dieu kien, ban xem lai ";
    getch();
    return 0;
}
```

☞ **Chương trình b cơ số c.**

```
// Tinh_Logarit_ b_coso_c.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int b,c;
float z;
main()
{
   clrscr();
   cout<<"Ban nhap so b= ";
```

```
    cin>>b;
    cout<<"Ban nhap vao co so c= ";
    cin>>c;
     if ((b>0)&&(c>0)&& c!=1))
         {
            z=log(b)/log(c);
            cout<<"Logarit co so "<<c<<"cua "<<b<<"= "<<z;
         }
     else
           cout<<"Khong thoa dieu kien, ban xem lai ";
    getch();
    return 0;
}
```

☞ **Chương trình a nhân b cơ số c.**

```
// Tinh_ anhanh_coso_c.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int a,b,c;
float z;
main()
{
    clrscr();
    cout<<"Ban nhap so a= ";
    cin>>a;
    cout<<"Ban nhap so b= ";
    cin>>b;
    cout<<"Ban nhap so c= ";
    cin>>c;
    if ((a>0)&&(b>0)&&(c>0)&&(c!=1))
        {
            z=log(a)/log(c)+log(b)/log(c);
            cout<<"Logarit co so "<<c<<"cua "<<a<<"* "<<b<<"= "<<z;
        }
    else
```

```
            cout<<"Khong thoa dieu kien, ban xem lai cua a, b, c";
        getch();
        return 0;
}
```

☞ **Chương trình a chia b cơ số c.**

```
// Tinh_ aChiab_coso_c.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int a,b,c;
float z ;
main()
{
    clrscr();
    cout<<"Ban nhap so a= ";
    cin>>a;
    cout<<"Ban nhap so b= ";
    cin>>b;
    cout<<"Ban nhap so c= ";
    cin>>c;
    if ((a>0)&&(b>0)&&(c>0)&&(c!=1))
        {
            z=((log(a)/log(c))-(log(b)/log(c)));
            cout<<"Logarit co so "<<c<<" cua "<<a<<"/"<<b<< "= "<<z;
        }
    else
            cout<<"Khong thoa dieu kien, ban xem lai cua a,b,c ";
    getch();
    return 0;
}
```

# ✍ Giải hệ phương trình tuyến tính có hai ẩn dùng ma trận

Hệ phương trình có dạng :

a11x + a12y = c1

a21x + a22y = c2

*Hướng dẫn:* Để giải hệ phương trình trên, đầu tiên bạn tính định thức :

a11 a12

a21 a22

Định thức = a11*a22–a21*a12

Định thức này phải khác 0

Thay cột các hệ số của ẩn x bằng cột của các số hạng tự do, tính định thức đó ta được kết quả là định thức x. Vậy:

c1 a12

c2 a22

Định thức x=c1*a22–c2*a12

Tương tự như vậy, nếu thay cột các hệ số của ẩn y bằng cột các số hạng tự do, tính định thức đó ta được kết quả là định thức y. Vậy:

a11 c1

a21 c2

Định thức y=c2*a11–c1*a21

x=Định thức x/Định thức

y=Định thức y/Định thức

**// HPTtuyentinh2an.cpp**

```
/* ax + by = m
   cx + dy = n */
#include <iostream.h>
#include <conio.h>
float a,b,c,d,m,n;
float dx,dy,dd;
main()
{
    clrscr();
    cout<<"a= ";
    cin>>a;
    cout<<"b= ";
    cin>>b;
    cout<<"c= ";
    cin>>c;
    cout<<"d= ";
```

```
    cin>>d;
    cout<<"m = ";
    cin>>m;
    cout<<"n= ";
    cin>>n;
    dd=a*d-b*c;
    dx=m*d-b*n;
    dy=a*n-c*m;
    if (dd==0)
        {
            if ((dx==0)&&(dy==0))
                cout<<"HPT vo so nghiem ";
            else
                cout<<"HPT vo nghiem ";
        }
    else
        {
            cout<<"x= "<<dx/dd<<"\n";
            cout<<"y= "<<dy/dd<<"\n";
        }
   getch();
   return 0;
}
```

## Giải hệ phương trình tuyến tính có ba ẩn dùng ma trận

Hệ phương trình có dạng:

a11x + a12y + a13z = c1
a21x + a22y + a23z = c2
a31x + a32y + a33z = c3

*Hướng dẫn*

Để giải hệ phương trình trên, đầu tiên bạn tính định thức:

a11 a12 a13
a21 a22 a23

a31 a32 a33

Định thức = a11*a22*a33+a12*a23*a31+a21*a32 *a13-a31*a22*a13-a21*a12*a33-a32*a23*a11

Định thức này phải khác 0

*Ghi chú:*

Để tính định thức cấp 3 này, các bạn nhân các phần tử trên đường chéo của định thức theo dấu huyền "\" cộng cho tích của 3 phần tử là 3 đỉnh của tam giác thứ nhất có cạnh đáy song song với đường chéo này, cộng cho tích của 3 phần tử là 3 đỉnh của tam giác thứ hai có cạnh đáy song song với đường chéo này*(để thấy rõ hơn, bạn có thể nhìn các phần tử mà chúng tôi đã dùng trong phép tính định thức ở trên, hai tam giác này ngược nhau).*

Bạn ghi nhận kết quả trên, sau đó bạn tính tương tự như vậy nhưng với đường chéo có dấu sắc "/" và với hai tam giác có cạnh đáy song song với đường chéo này. Ghi nhận kết quả.

*Sau đó, bạn lấy kết quả trước trừ cho kết quả sau, đó là kết quả của định thức. Ở trên chúng tôi bỏ dấu ngoặc, nên tất cả đều mang dấu trừ .*

Cách nhớ, bạn nhớ dấu huyền là cộng, dấu sắc là trừ

Để tính định thức x, Bạn thay cột các hệ số của ẩn x bằng cột của các số hạng tự do, tính định thức này ta được kết quả là định thức x. Vậy:

c1 a12 a13

c2 a22 a23

c3 a32 a33

Định thức x = c1*a22*a33+a12*a23*c3+c2*a32*a13-c3*a22*a13-c2*a12*a33-a32*a23*c1

Tương tự như vậy, nếu thay cột các hệ số của ẩn y bằng cột các số hạng tự do, tính định thức đó ta được kết quả là định thức y. Vậy :

a11 c1 a13

a21 c2 a23

a31 c3 a33

Định thức y = a11*c2*a33+c1*a23*a31+a21*c3*a13-a31*c2*a13-a21*c1*a33-c3*a23*a11

Tương tự như vậy, nếu thay cột các hệ số của ẩn z bằng cột các số hạng tự do, tính định thức đó ta được kết quả là định thức z. Vậy:

Định thức y = a11*a22*c3+a12*c2*a31+a21*a32*c1-a31*a22*c1-a21*a12*c3-a32*c2*a11

Vậy ta có thể tính x, y và z như sau:
x=Định thức x/Định thức
y=Định thức y/Định thức
z=Định thức z/Định thức

**// HPTTuyentinhbaan.cpp**

```
/* a11x + a12y + a13z = b1
   a21x + a22y + a23z = b2
   a31x + a32y + a33z = b3
   ax   +  by  +  cz  = o
   dx   +  ey  +  fz  = p
   gx   +  hy  +  iz  = q */
#include <iostream.h>
#include <conio.h>
float dd,dx,dy,dz;
float x,y,z;
float a,b,c,d,e,f,g,h,i;
float o,p,q;
main()
{
     // Nhap he so cua cac bien
     clrscr();
     cout<<"Ban nhap vao he so cua cac bien "<<"\n";
     cout<<"a= ";
     cin>>a;
     cout<<"b= ";
     cin>>b;
     cout<<"c= ";
     cin>>c;
     cout<<"d= ";
     cin>>d;
     cout<<"e= ";
     cin>>e;
     cout<<"f= ";
     cin>>f;
     cout<<"g= ";
     cin>>g;
```

```
        cout<<"h= ";
        cin>>h;
        cout<<"i= ";
        cin>>i;
        //Nhap he so cot tu do
        cout<<"Ban nhap vao he so cot tu do ";
        cout<<"o= ";
        cin>>o;
        cout<<"p= ";
        cin>>p;
        cout<<"q= ";
        cin>>q;
        dd=(a*e*i)+(b*f*g)+(d*h*c)-(g*e*c)-(d*b*i)-(h*f*a);
        dx=(o*e*i)+(b*f*q)+(p*h*c)-(q*e*c)-(p*b*i)-(h*f*o);
        dy=(a*p*i)+(o*f*g)+(d*q*c)-(g*p*c)-(d*o*i)-(q*f*a);
        dz=(a*e*q)+(b*p*g)+(d*h*o)-(g*e*o)-(d*b*q)-(h*p*a);
        if (dd!=0)
            {
                x=dx/dd;
                y=dy/dd;
                z=dz/dd;
                cout<<"Nghiem cua he phuong trinh la: x="<<x<<"   "<<
                            "y= "<<y<<" va z= "<< z;
            }
        else
            {
             if ((dx==0)&&(dy==0)&&(dz==0))
                cout<<"Vo so nghiem ";
             else
                cout<<"Vo nghiem ";
            }
        getch();
        return 0;
}
```

# ☎ Phần 3

# Bài tập sử dụng CASE ... OF

## ✍ Nhập năm dương lịch in ra năm âm lịch tương ứng

*Hướng dẫn*

Năm âm lịch là do CAN và CHI ghép lại, có tất cả là 10 CAN và 12 CHI sau đây:

| CAN | CHI |
|---|---|
| Giáp | Tí |
| Ất | Sửu |
| Bính | Dần |
| Đinh | Mẹo |
| Mậu | Thìn |
| Kỷ | Tỵ |
| Canh | Ngọ |
| Tân | Mùi |
| Nhâm | Thân |
| Quý | Dậu |
| | Tuất |
| | Hợi |

Can và Chi của năm âm lịch được lấy theo thứ tự lần lượt như trên, chẳng hạn năm 2000 là năm Canh Thìn thì năm 2001 là năm Tân tỵ Để được Can của một năm Y, ta căn cứ vào Ymod10 và Chi của một năm Y, ta căn cứ vào Ymod12

Ta có bảng của CAN và CHI tương ứng với số dư của Ymod10 (CAN) và Ymod12 (CHI) như sau:

| CAN (Ymod10) | Số dư Tương ứng | CHI (Ymod12) | Số dư Tương ứng |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Giáp | 4 | Tí | 4 |
| Ất | 5 | Sửu | 5 |
| Bính | 6 | Dần | 6 |
| Đinh | 7 | Mẹo | 7 |
| Mậu | 8 | Thìn | 8 |
| Kỷ | 9 | Tỵ | 9 |
| Canh | 0 | Ngọ | 10 |
| Tân | 1 | Mùi | 11 |
| Nhâm | 2 | Thân | 0 |
| Quý | 3 | Dậu | 1 |
| | | Tuất | 2 |
| | | Hợi | 3 |

Dùng Select Case Ymod10 để xác định trị của CAN.
Dùng Select Case Ymod12 để xác định trị của CHI.
Xuất năm âm lịch của Y là CAN+CHI.

```
// Nam_am_lich.cpp
#include <iostream.h>
#include <conio.h>
#include <string.h>
int nam;
char can[5];
char chi[5];
main()
{
    clrscr();
    cout<<"Ban nhap vao nam duong lich ";
    cin>>nam;
    switch (nam%10)
    {
      case 4: strcpy(can,"Giap");break;
      case 5: strcpy(can,"At");break;
      case 6: strcpy(can,"Binh");break;
      case 7: strcpy(can,"Dinh");break;
      case 8: strcpy(can,"Mau");break;
```

```
        case 9: strcpy(can,"Ky");break;
        case 0: strcpy(can,"Canh");break;
        case 1: strcpy(can,"Tan");break;
        case 2: strcpy(can,"Nham");break;
        case 3: strcpy(can,"Quy");break;
    }
    switch (nam % 12 )
    {
        case 4: strcpy(chi,"Ti");break;
        case 5: strcpy(chi,"Suu");break;
        case 6: strcpy(chi,"Dan");break;
        case 7: strcpy(chi,"Meo");break;
        case 8: strcpy(chi,"Thin");break;
        case 9: strcpy(chi,"Ti");break;
        case 10:strcpy(chi,"Ngo");break;
        case 11: strcpy(chi,"Mui");break;
        case 0: strcpy(chi,"Than");break;
        case 1: strcpy(chi,"Dau");break;
        case 2: strcpy(chi,"Tuat");break;
        case 3: strcpy(chi,"Hoi");break;
    }
    cout<<"Ung voi nam duong lich  "<<nam<<'\n'<<
                    "Nam am lich se la: "<<can<<" "<<chi;
    getch();
    return 0;
}
```

⇨ Trong chương trình này, các bạn vẫn dùng cấu trúc switch (*giá trị*) như bài trước.

⇨ Tuy nhiên, trong bài này các bạn dùng hàm strcpy (biến, chuỗi). *(bạn lưu ý chuỗi không được có khoảng trắng, các kí tự phải viết liên tục với nhau). Hàm này sẽ gán chuỗi cho biến. Vì phải dùng hàm strcpy()*, các bạn phải khai báo tập tin tiêu đề string.h, vì tập tin tiêu đề này chứa hàm strcpy.

⇨ Trong phát biểu switch bạn muốn gán các chuỗi Giáp, Ất...Quý cho biến can tương ứng với giá trị đúng của phép chia số dư, bạn phải dùng

hàm strcpy (biến, chuỗi) để gán cho biến can, bạn không thể gán can = "Giáp" như các ngôn ngữ khác. Bạn cần lưu ý ở điểm này. Tương tự như vậy với phát biểu switch sau.

⇨ Sau khi gán xong cho hai biến can và biến chi, ta cho xuất ra màn hình năm âm lịch ứng với năm dương lịch mà người sử dụng nhập vào.

⊙ Bạn chạy thử chương trình, sau đó nhập vào năm dương lịch là 2000, bạn sẽ có kết quả như sau:
Ứng với năm dương lịch 2000
Năm âm lịch sẽ là Canh Thìn.

⊙ Chạy lại chương trình và nhập vào năm 2001, bạn sẽ có kết quả như sau:
Ứng với năm dương lịch 2001
Năm âm lịch sẽ là Tân tỵ.

# ☎ Phần 4

## Bài tập sử dụng các phát biểu lặp FOR, WHILE

### ✍ Tìm một số có 4 chữ số, sao cho tổng các lũy thừa 4 của bốn chữ số thì bằng chính nó

$$\overline{abcd}=1000a+100b+10c+d=a^4+b^4+c^4+d^4$$

Bài này nếu tự mình tính thì rất khó, tuy nhiên đối với máy tính thì dễ dàng, không cần đến một giây, nó sẽ thử được hết tất cả các trường hợp và cho ta kết quả, tuy nhiên con người phải lập trình cho nó. Bài này cũng rất đơn giản, bạn chỉ cần dùng bốn vòng For lồng nhau với chỉ số chạy thích hợp thì nó sẽ cho ra đúng kết quả.

Các bạn cần chú ý là một số thì chữ số đầu tiên phải khác 0. Không thể có số 0237, nếu có số như thế, người ta chỉ cần ghi 237. Bạn cần lưu ý điều này để lập trình cho đúng.

☞ Cách 1

```
// Tinhtoan.cp
#include <iostream.h>
#include <conio.h>
int a,b,c,d;
main()
{
    clrscr();
    for (a=1; a<=9; a++)
        for (b=0; b<=9; b++)
         for (c=0; c<=9; c++)
            for (d=0; d<=9; d++)
              if ((a*a*a*a)+(b*b*b*b)+(c*c*c*c)+(d*d*d*d)==
```

```
                                        1000*a+100*b+10*c+d)
                    cout<<a<<b<<c<<d<<'\n';
            getch();
            return(0);
}
```

☞ Cách 2

```
// Tinhtoan.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int a,b,c,d;
main()
{
    clrscr();
    for (a=1; a<=9; a++)
      for (b=0; b<=9; b++)
        for (c=0; c<=9; c++)
          for (d=0; d<=9; d++)
            if (pow(a,4)+pow(b,4)+pow(c,4)+pow(d,4)==
                                    1000*a+100*b+10*c+d)
                    cout<<a<<b<<c<<d<<'\n';
            getch();
            return(0);
}
```

⇨ Bạn dùng 4 vòng For lồng nhau để tìm ra được các số thoả yêu cầu.

- Cách 1: bạn tính $a^4$ bằng cách (a*a*a*a)
- Cách 2: bạn tính bằng cách dùng hàm pow(a,x) tức là tính a lũy thừa x. Khi dùng hàm này ta phải khai báo tập tin tiêu đề math.h vì tập tin này chứa hàm pow(a,x).

⊙ Chạy thử chương trình, bạn sẽ có kết quả như sau:

```
1634
8208
9474
```

## ✍ Tính:

- ❑ n! (n giai thừa)
- ❑ $S=1^2+2^2+3^2+.....+n^2$
- ❑ $S=1^3+2^3+3^3+.....+n^3$
- ❑ $S=1/1^2+1/2^2+1/3^2+...+1/n^2$

```
// Giaithua.cpp
#include <iostream.h>
#include <conio.h>
int i;
int n;
long int ketqua;
main()
{
    clrscr();
    cout<<"Ban nhap vao so de tinh giai thua ";
    cin>>n;
    if (n==0)
        cout<<"0!=1";
    else
        ketqua=1;
          for (i=1; i<=n; i++)
              ketqua=(ketqua*i);
        cout<<"n!= "<<ketqua;
    getch ();
    return 0;
}
// Tongbinhphuong.cpp
#include <iostream.h>
#include <conio.h>
int i;
int n;
long int ketqua;
main()
{
```

```
    clrscr();
    cout<<"Ban nhap vao so n ";
    cin>>n;
    for (i=1; i<=n; i++)
        ketqua=ketqua+i*i;
    cout<<"Voi n= "<<n<<", ta co tong binh phuong = "<<ketqua;
    getch();
    return 0;
}
// TongLapphuong.cpp
#include <iostream.h>
#include <conio.h>
int i;
int n;
long int ketqua;
main()
{
    clrscr();
    cout<<"Ban nhap vao so n ";
    cin>>n;
    for (i=1; i<=n; i++)
        ketqua=ketqua+i*i*i;
    cout<<"Voi n= "<<n<<", ta co tong lap phuong = "<<ketqua;
    getch();
    return 0;
}
// Tongbinhphuongmottren.cpp
#include <iostream.h>
#include <conio.h>
float i;
int n;
float ketqua;
main()
{
    clrscr();
    cout<<"Ban nhap vao so n ";
```

```
    cin>>n;
    for (i=1; i<=n; i++)
        ketqua=ketqua+1/(i*i);
    cout<<"Voi n= "<<n<<", ta co tong binh phuong mot tren= "<<ketqua;
    getch();
    return 0;
}
```

- Bạn nhấn Ctrl+F9 để chạy thử chương trình
- Nhập n=3, bạn sẽ có thông báo là 3!=6
- Kiểm lại ta thấy: 3!=1x2x3=6
- Bạn chạy chương trình và nhập n=3, bạn sẽ được thông báo kết quả: Với n=3, ta có tổng bình phương=14
- Kiểm tra lại ta thấy: 1+4+9=14
- Bạn chạy chương trình và nhập n=3, bạn sẽ được thông báo kết quả: Với n=3 ta có tổng lập phương=36.
- Kiểm tra lại ta thấy: 1+8+27=36
- Bạn chạy chương trình và nhập n=3, bạn sẽ được thông báo kết quả: Với n=3 ta có tổng bình phương một trên =1.36
- Kiểm tra lại ta thấy: 1/1+1/4+1/9=1.36

## In ra các số nguyên tố từ 2 đến n

Như bạn đã biết số nguyên tố nhỏ nhất là 2 (*đó là số nguyên tố chẵn duy nhất)* và theo định nghĩa số nguyên tố là số lớn hơn 1, chỉ có hai ước số là 1 và chính nó.

Các bạn có thể dùng nhiều thuật toán để giải quyết bài toán này. Trong sách Toán phổ thông đã có thuật toán sàng Erathostene, các bạn có thể tham khảo để thực hiện. Ở đây chúng tôi dùng cách khác để tìm ra các số nguyên tố từ 2 đến một số nào đó.

*Một số N là một số nguyên tố nếu như "Tất cả các số nguyên kể từ 2 cho đến căn bậc hai của số này (số N) không phải là ước số của số này".*

```
// Nguyento.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
```

```
int  k,i,j,n;
int  Max;
main()
{
    clrscr();
    cout<<"Ban nhap vao so de xem so nguyen to den so nay: ";
    cin>>Max;
    if (Max<2)
      cout<<"Khong co so nao ca, vi so ban nhap nho hon 2 ";
    else
      j=0;
      for (n=2; n<=Max; n++)
          {
            k=floor(sqrt(n));
            i=2;
            while ((n%i!=0)&&(i<=k))
              i=i+1;
            if (i>k)
              {
                  if (j%10==0)
                      cout<<'\n';
                  j=j+1;
                  cout<<n<<"  ";
              }
          }
    getch();
    return 0;
}
```

⇨ Trong chương trình trên, nếu bạn nhập vào một số nhỏ hơn 2, chương trình sẽ báo là không có số nguyên tố nào cả.

⇨ Ngược lại sẽ in ra màn hình, các số nguyên tố từ 2 đến số mà bạn nhập vào.

⇨ Biến j là biến điều khiển trình bày màn hình.

⇨ Hàm sqrt(x) để tính căn bậc hai của x.

⇨ Hàm Floor(Num) cho ta phần nguyên của Num.

⊙ Nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Giả sử bạn nhập Max=1

⊙ Chương trình sẽ thông báo là không có số nguyên tố nào cả.

⊙ Bạn chạy lại chương trình, sau đó nhập vào số Max=45, bạn sẽ thấy thông báo trên màn hình các số nguyên tố như sau:

2 3 5 7 11 13 17 19 23 29 31 37 41 43

⊙ Bạn thử nhập thêm một số trường hợp nữa để kiểm tra nhé!

# Xác định một số có phải là số nguyên tố

Trong chương trình trên, chúng ta nhập vào một số, chương trình sẽ cho in ra các số nguyên tố có từ 2 cho đến số nhập vào. Trong chương trình này, chúng tôi yêu cầu bạn viết chương trình sao cho khi người sử dụng nhập một số nguyên bất kỳ, bạn viết code như thế nào để khi nhập vào một số nguyên bất kỳ thì nó sẽ báo chính xác đúng với giá trị của số đó, và cho biết số đó có phải là một số nguyên tố hay không.

```
// Hoisonguyento.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int n,i;
float m;
main()
{
    clrscr();
    cout<<"Ban nhap vao so n nguyen duong";
    cin>>n;
    m=sqrt(abs(n));
    i=2;
    while ((n%i!=0)&&(i<=n))
        i=i+1;
    if ((i>m)&&(n>=2))
        cout<<"So "<<n<<"la mot so nguyen to.";
    else
     cout<<"So "<<n<<"Khong phai la mot so nguyen to";
```

```
    getch();
    return 0;
}
```

⇨ Hàm abs() để lấy trị tuyệt đối của đối số.

⇨ Bạn tự tìm hiểu chương trình.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Bạn nhập thử n=43, bạn sẽ được thông báo
Số 43 là một số nguyên tố.

⊙ Bạn chạy lại và nhập n=100, bạn sẽ thấy thông báo
Số 100 không phải là một số nguyên tố.

# Đổi số nhị phân, bát phân thập lục phân sang số thập phân

⇨ Người sử dụng nhập vào một số nhị phân, số bát phân, số thập lục phân. Nếu nhập sai, chương trình sẽ báo để nhập lại.

⇨ Để viết được chương trình này, các bạn cần phải nắm đầy đủ kiến thức cơ bản về số, đồng thời với các lý luận để viết code cho chương trình.

⇨ Thông thường, chúng ta đang dùng hệ thập phân để biểu diễn các số nguyên. Số thập phân có cơ số là 10 gồm 10 kí số từ 0 đến 9.

*Ví dụ:*

◎ 456 được dùng để kí hiệu: $4.10^2+5.10^1+6.10^0$

◎ Bạn biết một số có số mũ bằng 0 thì sẽ bằng 1.

◎ Do đó 456=400+50+6

◎ Số 456 trong đó số 6 (số tận cùng bên phải) sẽ có trọng số là 0, số 5 có trọng số là 1, số 4 có trọng số là 2. Sau này đối các hệ có cơ số khác thì cách tìm trọng số cũng như vậy.

⇨ Tuy nhiên trong nhiều trường hợp, việc dùng các cơ số khác 10 sẽ thuận tiện hơn. Đặc biệt trong máy tính thường dùng hệ nhị phân (với cơ số là 2) tức là chỉ các kí số 0 hoặc 1. Ngoài ra trong kỹ thuật máy tính người ta còn dùng hệ bát phân (có cơ số là 8) dùng các kí số từ 0 đến 7, hệ thập lục phân (có cơ số 16) dùng các kí số từ 0 đến 9, và các chữ A, B, C, D, E, F. Bạn có thể đối chiếu qua bảng sau:

## Bảng các số từ 0 đến 15 của bốn hệ

| Thập phân | Nhị phân | Bát phân | Thập lục phân |
|---|---|---|---|
| 0 | 0000 | 0 | 0 |
| 1 | 0001 | 1 | 1 |
| 2 | 0010 | 2 | 2 |
| 3 | 0011 | 3 | 3 |
| 4 | 0100 | 4 | 4 |
| 5 | 0101 | 5 | 5 |
| 6 | 0110 | 6 | 6 |
| 7 | 0111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |

⇨ Để đổi một số từ một hệ nào đó sang hệ thập phân, các bạn tính tổng của từng kí số nhân với cơ số của hệ đó với số mũ là trọng số của kí số đó.

*Ví dụ:*

⇨ Để đổi 1111 ra số thập phân, bạn sẽ tính như sau:

$(1111)_2=1x2^3+1x2^2+1x2^1+1x2^0$

$(1111)_2=8+4+2+1=(15)_{10}$

⇨ Đổi $(10001)_2$ ra số thập phân

$(10001)_2=1x2^4+0x2^3+0x2^2+0x2^1+1x2^0$

$(10001)_2=16+0+0+0+1=(17)_{10}$

⇨ Đổi $(245)_8$ ra số thập phân

$(245)_8=2x8^2+4x8^1+5x8^0$

$(245)_8=128+32+5=(165)_{10}$

⇨ Đổi $(2AE0B)_{16}$ ra số thập phân

$(2AE0B)_{16}=2x16^4+Ax16^3+Ex16^2+Bx16^0$

$(2AE0B)_{16}=131072+40960+3584+11=(175627)_{10}$

**// Nhiphan_sang_Thapphan.cpp**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
char y;
int i,k,x,n,e,m;
float Tong;
int Sonhiphan[20];
main()
{
    clrscr();
    cout << "Ban nhap vao chieu dai so nhi phan ";
    cin>>m;
    cout<<"Ban nhap vao cac ki so nhi phan ";
    for (i=1;i<=m;i++)
      cin>>Sonhiphan[i];
    i=1;
    k=1;
    while (k<=m)
        {
            n=Sonhiphan[i];
            if ((n==1) || (n==0))
                {
                    Tong=Tong+n*pow(2, m-i);
                    i=i+1;
                    k=k+1;
                }
            else
                {
                    cout<<"So ban nhap khong phai la so nhi phan.";
                    k=m+1;
                    goto y;
                }
```

```
        }
        cout<<"So nhi phan ban nhap la: ";
        for (i=1; i<=m; i++)
          cout<<Sonhiphan[i]<<"  ";
        cout<<"\n So nhi phan tren doi sang so thap phan la: "<<Tong;
        y ;;
    getch();
    return 0;
}
```

**// Batphan_sang_Thapphan.cpp**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
char y;
int i,k,x,n,e,m;
float Tong;
int Sobatphan[20];
main()
{
    clrscr();
    cout<<"Ban nhap vao chieu dai so bat phan ";
    cin>>m;
    cout<<"Ban nhap vao cac ki so bat phan ";
    for (i=1; i<=m; i++)
      cin>>Sobatphan[i];
    i=1;
    k=1;
    while (k<=m)
        {
            n=Sobatphan[i];
            if ((n==0)||(n==1)||(n==2)||(n==3)||(n==4)||(n==5)||
              (n==6)||(n==7))
                {
                    Tong=Tong+n*pow(8, m-i);
                    i=i+1;
                    k=k+1;
```

```
            }
        else
            {
                cout<<"So ban nhap khong phai la so bat phan.";
                k=m+1;
                goto y;
            }
    }
    cout<<"So bat phan ban nhap la: ";
    for (i=1; i<=m; i++)
      cout<<Sobatphan[i]<<"  ";
      cout<<"\n So bat phan tren doi sang so thap phan la: "
            << Tong;
    y :;
  getch();
  return 0;
}
```

⇨ Trong C++ chúng tôi không viết chương trình đổi một số thập lục phân sang số thập phân, các bạn tự tìm hiểu để viết thử.

⇨ Trong hai chương trình đổi số nhị phân và bát phân sang số thập phân, bạn khai báo một mảng kiểu số nguyên để chứa các kí số của chúng. Sau khi nhập xong, bạn lấy ra từng kí tự tương ứng từ trái sang phải bằng vòng lặp while. Nếu một kí số bất kì trong số nhị phân không phải là số 0 hoặc số 1, nếu một kí số bất kỳ trong số bát phân không phải từ kí số 0 đến số 7 thì sẽ thông báo không phải là số nhị phân hoặc bát phân, gán biến k=m+1 để thoát khỏi vòng lặp. Nếu bạn không chú ý chỗ này, vòng lặp sẽ thực hiện mãi. Lệnh Goto để chuyển đến vị trí cần thiết.

⇨ Bạn chú ý cách dùng các biểu thức để lấy đúng kết quả.

# Đổi số nhị phân, thập phân thập lục phân sang số bát phân

☞ Yêu cầu cũng giống như bài trên, chỉ khác là đổi sang số bát phân.

⇨ Trong bài này, các bạn cần phải biết muốn đổi một số thập phân sang số bát phân, bạn thực hiện như sau:

⇨ Bạn lấy số thập phân chia cho 8 được thương và số dư. Ghi nhận số dư đó.

⇨ Lấy thương ở trên tiếp tục chia cho 8, được một thương mới và số dư. Ghi nhận số dư đó.

⇨ Cứ tiếp tục như vậy cho đến khi thương tìm được chia 8 bằng 0. Ghi nhận số dư.

⇨ Lấy số dư theo chiều ngược lại, sẽ cho ta kết quả của số bát phân cần tìm.

*Ví dụ:*

Đổi số thập phân 12345 sang số bát phân.

Trước hết chia 12345 cho cơ số 8 ta được:

12345=8x1543+1 (số dư là 1, thương là 1543)

1543=8x192+7 (số dư là 7, thương là 192)

192=8x24+0 (số dư là 0, thương là 24)

24=8x3+0 (số dư là 0, thương là 3)

3=8x0+3 (số dư là 3, thương là 0)

Bạn thấy thương đã bằng 0, quá trình dừng lại.

⇨ Vậy các số dư chính là các chữ số của khai triển cơ số 8 của 12345, ta suy ra rằng:

$$(12345)_{10}=(30071)_8$$

⇨ Để đổi một số nhị phân, số thập lục phân sang số bát phân, bạn đổi chúng sang số thập phân, sau đó đổi số thập phân này sang số bát phân như trên. Bạn phải qua một bước trung gian.

⇨ Trong chương trình này, chúng tôi xem như người nhập phải nhập đúng các loại số, chúng tôi bỏ qua phần kiểm tra.

⇨ Chúng tôi không viết phần đổi số thập lục phân sang số bát phân trong C++.

**// Thapphan_sang_Batphan.cpp**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
int i,n,q,m;
float Tong;
int a[10];
```

```
main()
{
    clrscr();
    cout<<"Ban nhap vao so thap phan ";
    cin>>q;
    m=q;
    i=1;
    while (q!=0)
        {
          a[i]=q%8;
          q=q/8;
          i=i+1;
          n=n+1;
        }
        if (m==0)
          cout<<"So thap phan 0 doi sang so bat phan la: 0";
        else
        cout<<"So thap phan "<<m<<"doi sang so bat phan la: ";
        for (i=n; i>=1; i--)
          cout<<a[i]<<"  ";
     getch();
     return 0;
}
```

**// Nhiphan_sang_Batphan.cpp**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
int i,j,k,n,e,m,u,v,x,q;
int Tong;
int a[10];
int b[20];
main()
{
    clrscr();
    cout<<"Ban nhap vao chieu dai so nhi phan ";
    cin>>m;
```

```
    cout<<"Ban nhap cac ki so nhi phan ";
    for (i=1; i<=m; i++)
        cin>>a[i];
    j=1;
    k=1;
    while (k<=m)
        {
            n=a[j];
            Tong=Tong+n*pow(2,m-j);
            j=j+1;
            k=k+1;
        }
    q=Tong;
    u=1;
    while (q!=0)
        {
          b[u]=q%8;
          q=q/8;
          u=u+1;
          v=v+1;
        }
        cout<<"So nhi phan: ";
        for (i=1; i<=m; i++)
          cout<<a[i]<<"  ";
        cout<<"\n Doi sang so bat phan la: " ;
        for (i=v; i>=1; i --)
          cout<<b[i]<<" ";
      getch();
      return 0;
}
```

◎ Chạy thử chương trình đổi số thập phân sang số bát phân. Giả sử bạn nhập vào số thập phân là 45, bạn sẽ có kết quả như sau:
Số thập phân 45 đổi sang số bát phân là 55.
Bạn kiểm tra lại xem có đúng không.

◎ Chạy thử chương trình đổi số nhị phân sang số bát phân. Giả sử bạn nhập vào số nhị phân 1111, bạn sẽ có kết quả như sau:

Số nhị phân 1111 đổi sang số bát phân là 17.

## ✍ Đổi số bát phân, thập phân, thập lục phân sang số nhị phân

☞ Yêu cầu cũng giống như bài trên, chỉ khác là đổi sang số nhị phân.

⇨ Trong bài này, các bạn cần phải biết muốn đổi một số thập phân sang số nhị phân, bạn thực hiện như sau:

⇨ Bạn lấy số thập phân chia cho 2 được thương và số dư. Ghi nhận số dư đó.

⇨ Lấy thương ở trên tiếp tục chia cho 2, được một thương mới và số dư. Ghi nhận số dư đó.

⇨ Cứ tiếp tục như vậy cho đến khi thương tìm được chia 2 bằng 0. Ghi nhận số dư.

⇨ Lấy số dư theo chiều ngược lại, sẽ cho ta kết quả của số nhị phân cần tìm.

*Ví dụ:*

Đổi số thập phân 45 sang số nhị phân.

Trước hết chia 45 cho cơ số 2 ta được:

45 = 2x22+1 (số dư là 1, thương là 22)
22 = 2x11+0 (số dư là 0, thương là 11)
11=2x5+1 (số dư là 1, thương là 5)
5=2x2+1 (số dư là 1, thương là 2)
2=2x1+0 (số dư là 0, thương là 1)
1=2x0+1 (số dư là 1, thương là 0)

Bạn thấy thương đã bằng 0, quá trình dừng lại.

⇨ Vậy các số dư chính là các chữ số của khai triển cơ số 2 của 45, ta suy ra rằng:

$$(45)_{10}=(1\ 0\ 1\ 1\ 0\ 1)_2$$

⇨ Kiểm tra lại, bạn thấy:

$$(1\ 0\ 1\ 1\ 0\ 1)_2=2^5+2^3+2^2+1$$
$$=32+8+4+1=(45)_{10}$$

⇨ Trong chương trình này, chúng tôi không viết đổi số thập lục phân sang số nhị phân. Các bạn tự viết và xem đây như là bài tập.

```
// Thapphan_sang_Nhiphan.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int i,n,q,e,m;
float Tong;
int a[10];
main()
{
    clrscr();
    cout<<"Ban nhap vao so thap phan ";
    cin>>q;
    m=q;
    i=1;
    while (q!=0)
        {
            a[i]=q%2;
            q=q/2;
            i=i+1;
            n=n+1;
        }
        cout<<"So thap phan "<<m<<"doi sang so nhi phan la: ";
        for (i=n; i>=1; i--)
            cout<<a[i]<<"  ";
    getch();
    return 0;
}

// Batphan_sang_Nhiphan.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int i,j,k,n,e,m,u,v,q;
```

```
float Tong;
int a[10];
int b[20];
main()
{
    clrscr();
    cout<<"Ban nhap chieu dai so bat phan ";
    cin>>m;
    cout<<"Nhap cac ki so bat phan: ";
    for (i=1; i<=m; i++)
        cin>>a[i];
    i=1;
    k=1;
    while (k<=m)
        {
            n=a[i];
            Tong=Tong+n*pow(8, m-i);
            i=i+1;
            k=k+1;
        }
   q=Tong;
   u=1;
   while (q!=0)
       {
         b[u]=q%2;
         q=q/2;
         u=u+1;
         v=v+1;
       }
      cout<<"So bat phan: ";
      for (i=1; i<=m; i++)
          cout<<a[i]<<" ";

      cout<<"\n Doi sang so nhi phan la: ";
      for (i = v; >=1; i--)
        cout<<b[i]<<"  ";
```

```
    getch();
    return 0;
}
```

◎ Chạy thử chương trình đổi số thập phân sang số nhị phân. Giả sử bạn nhập vào số thập phân là 15, bạn sẽ có thông báo như sau:
Số thập phân 15 đổi ra số nhị phân là 1111.

◎ Chạy thử chương trình đổi số bát phân sang số nhị phân. Giả sử bạn nhập vào số bát phân là 17, bạn sẽ có thông báo như sau:
Số bát phân 17 đổi ra số nhị phân là 1111.

# Tính tổng hai số thập phân nguyên dương có cùng chiều dài

⇨ Giả sử rằng khai triển thập phân của a và b là:

$$a=(a_{n-1}\ a_{n-2}...a_1\ a_0)_{10}$$

$$b=(b_{n-1}\ b_{n-2}....b_1\ b_0)_{10}$$

Sao cho a và b đều có n bit

⇨ Để cộng a và b, trước hết cộng hai bit tận cùng bên phải của chúng, tức là:

$$a_0+b_0 = C_0 10+S_0$$

⊙ Ở đây $S_0$ là bit phải tận cùng trong khai triển thập phân của a+b

⊙ $C_0$ là số nhớ (nó có thể bằng 0 hoặc 1)

⊙ Kế tiếp chúng ta cộng hai bit kế tiếp và số nhớ.

$$a_1+b_1+C_0=C_1 10+S_1$$

⊙ Ở đây $S_1$ là bit tiếp theo (tính từ bên phải) trong khai triển thập phân a+b và $C_1$ là số nhớ. Tiếp tục quá trình như vậy bằng cách cộng bit tương ứng trong hai khai triển thập phân và số nhớ để xác định bit tiếp sau tính từ bên phải trong khai triển thập phân của tổng a+b. Ở giai đoạn cuối cùng, cộng $a_{n-1}$ và $b_{n-1}$ để nhận được $C_{n-1}10+S_{n-1}$. Bit đứng đầu của tổng là $S_n=C_{n-1}$

⇨ Kết quả thủ tục này tạo ra được khai triển thập phân của tổng, cụ thể là $a+b=(S_n\ S_{n-1}\ S_{n-2}\ ...\ S_1\ S_0)_{10}$

**// Conghaiso_Thapphan.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,k,n,m,s,e,d,c;
int a[20];
int b[20];
int x[20];
int y[20];
int kq[21];
main()
{
    clrscr();
    // Nhap vao hai so thap phan
    cout<<"Ban nhap vao chieu dai cua hai so thap phan: ";
    cin>>m;
    cout<<"Ban nhap vao so thap phan thu nhat: ";
    for (i=1; i<=m; i++)
        cin>>a[i];
    cout<<"Ban nhap vao so thap phan thu hai: ";
    for (i=1; i<=m; i++)
        cin>>b[i];
  // Lay tung ki tu tu phai qua trai cho vao mang x
    for (i=1;i<=m;i++)
        {
            n = a[m+1-i];
            x[i]=n;
        }
  // Lay tung ki tu tu phai qua trai cho vao mang y
    for (i=1; i<=m; i++)
        {
            s=b[m+1-i];
            y[i]=s;
        }
  // Tinh toan cong hai so thap phan, cho vao mang kq
  c=0;
  for (k=1; k<=m; k++)
      {
```

```
        d=(x[k]+y[k]+c)/10;
        kq[k]=(x[k]+y[k]+c)-(10*d);
        c=d;
    }
    kq[k]=d; // Cho phan tu nho vao vi tri k+1
    // In ra man hinh so thap phan thu nhat
    cout<<"So thap phan thu nhat la:   ";
    for (i=m; i>=1; i--)
        cout<<x[i]<<"   ";
    cout<<"\n";
    // In ra man hinh so thap phan thu hai
    cout<<"So thap phan thu hai la:  ";
    for (i=m; i>=1; i--)
        cout<<y[i]<<"   ";
    cout<<"\n";
    // In ra ket qua la tong cua hai so thap phan
    cout<<"Tong cua hai so nhi phan tren la: ";
    if (d==1)
      for (i=m+1; i>=1; i--)
          cout<<kq[i]<<"   ";
    else
     cout<<"  ";
      for (i=m; i>=1; i--)
          cout<<kq[i]<<"  ";
    getch();
    return 0;
}
```

⇨ Bạn nhấn Ctrl+F9 để chạy thử chương trình

⇨ Bạn nhập số thập phân thứ nhất là 222
số thập phân thứ hai là 333
Bạn sẽ có kết quả là 555
(không có số 0 ở trước)
Nếu bạn nhập số thứ nhất là 56
số thứ hai là 59
Kết quả là 115 (có số nhớ 1)

# ☎ Phần 5

# Bài tập sử dụng MẢNG

## ✍ Tìm vị trí đầu tiên của số lớn nhất, số nhỏ nhất trong mảng có 10 phần tử

Trong chương trình này, các bạn sẽ dùng đến thuật toán tìm vị trí đầu tiên mà bạn gặp số nguyên lớn nhất và số nguyên nhỏ nhất, trong đó các số này không nhất thiết phải khác nhau, thuật toán này cũng giống như thuật toán tìm số lớn nhất, nhỏ nhất trong dãy. Đối với yêu cầu của bài, các bạn cần phải xác định được vị trí khi gặp đầu tiên.

Trong chương trình này, chúng tôi thiết lập cho người sử dụng nhập một dãy gồm 10 phần tử, sau khi nhập xong, bạn sẽ biết vị trí đầu tiên của số lớn nhất và vị trí đầu tiên của số nhỏ nhất.

```
// SoMax.cpp
#include <iostream.h>
#include <conio.h>
int i,j,n,Max,Vitri1,Vitri2;
int A[10];
main()
{
    clrscr();
    cout<<"Ban nhap vao 10 phan tu cua mang. \n";
    n=10;
    for (i=1; i<=n; i++)
        {
            cout<<"Ban nhap vao phan tu A["<<i<<"]= ";
            cin>>A[i];
        }
    for (i=1; i<=n; i++)
            if (Max<A[i])
```

```
        {
            Max=A[i];
            Vitri1=i;
        }
    cout<<"Vi tri cua so lon nhat gap dau tien la: "<<Vitri1<<'\n';
    for (j=1; j<=n; j++)
      if (Max>A[j])
        {
            Max=A[j];
            Vitri2=j;
        }
    cout<<"Vi tri cua so nho nhat gap dau tien la:  "<<Vitri2;
     getch();
     return 0;
}
```

Chương trình này cũng khá đơn giản, các bạn có thể tự tìm hiểu.

- Bạn nhấn Ctrl+F9 để chạy thử chương trình.
- Giả sử bạn nhập vào 10 phần tử như sau:

  A[1]=1
  A[2]=2
  A[3]=3
  A[4]=4
  A[5]=5
  A[6]=6
  A[7]=7
  A[8]=8
  A[9]=4
  A[10]=0

- Bạn sẽ thấy thông báo kết quả như sau:

  Vị gặp đầu tiên của số lớn nhất là: 8
  Vị trí gặp đầu tiên của số nhỏ nhất là: 10

## ✍ Nhập hai số nguyên nhỏ trước, lớn sau. Trong khoảng hai số này, tính:

- ❑ Các số chẵn có trong dãy
- ❑ Các số lẻ có trong dãy
- ❑ Tổng các số chẵn có trong dãy
- ❑ Tổng các số lẻ có trong dãy
- ❑ Tổng các số có trong dãy

```
// Dayso.cpp
#include <iostream.h>
#include <conio.h>
int chan[50];
int le[50];
int i, j;
int m, n;
int k;
int q;
long int Tong1,Tong2,Tong3;
main()
{
   // Nhap day so tu m den n
   cout<<"Ban tinh tu so:  " ;
   cin>>m;
   cout<<"\n Den so:  ";
   cin>>n;
   // Nhap so chan vao mang theo thu tu cua k
   k=0;
   for (i=m; i<=n; i++)
         if ((i%2)==0)
            {
```

```
            k=k+1;
            chan[k]=i;
          }
 // In ra man hinh cac so chan tu m den n
cout<<"Cac so chan tu  "<<m<<" den "<<n<<" la: ";
for (i=1; i<=k; i++)
      cout<<chan[i]<<"   ";
//  Nhap vao mang cac so le theo thu tu cua q
q=0;
for (j=m; j<=n; j++)
        if ((j%2)==1)
          {
            q=q+1;
            le[q]=j;
          }
cout<<"\n";
// In ra man hinh cac so le
cout<<"Cac so le tu "<<m<<" den "<<n<< " la: ";
for (j=1; j<=q; j++)
   cout<<le[j]<<"  ";
 // Tinh tong cac so chan
cout<<"\n";
Tong1=0;
for (i=1; i<=k; i++)
   Tong1=Tong1+chan[i];
cout<<"Tong cac so chan tu m den n la:  "<<Tong1;
// Tinh tong cac so le
cout<<"\n";
Tong2=0;
for (i=1; i<=q; i++)
   Tong2=Tong2+le[i];
cout<<"Tong cac so le tu m den n la:  "<<Tong2;
 // Tinh tong cac so chan va so le
cout << "\n";
Tong3 = Tong1 + Tong2;
cout << " Tong cac so chan va cac so le tu m den n la:  "<< Tong3;
```

```
    getch();
    return 0;
}
```

⇨ Trong chương trình trên, chúng tôi khai báo hai mảng một chiều có tên chan và le có 50 phần tử, các mảng này có kiểu nguyên.

⇨ Khai báo hai biến i và j có kiểu nguyên, dùng làm biến chạy trong các vòng lặp for. Trong C++ vòng lặp for có thể chạy từ số nhỏ đến số lớn như:

```
for (j=1; j<=100; j++)
```

⇨ Ngoài ra nó còn có thể chạy từ số lớn đến số nhỏ theo câu lệnh sau

```
for (j=50; j>=1; j--)
```

⇨ Đây là một vòng lặp xác định, nếu bạn biết chắc lặp từ số nào đến số nào, bạn dùng vòng lặp for.

⇨ Sau khi thoát khỏi vòng lặp, giá trị của j trong câu lệnh trước sẽ bằng 101. Câu lệnh sau j=0.

⇨ Khai báo biến k và q có kiểu nguyên, hai biến này dùng để làm chỉ số của mảng khi gán các giá trị, đồng thời giá trị của biến k cho ta biết có bao nhiêu số chẵn có trong dãy từ m đến n, tương tự như vậy, biến q cho ta biết trường hợp của số lẻ. Các biến này chạy từ 1 đến một số là số phần tử chẵn, số phần tử lẻ có trong dãy.

⇨ Ba biến Tong1, Tong2, Tong3 có kiểu long int, đây là kiểu số nguyên dài (*Giá trị lớn hơn số nguyên Integer, kiểu này có giá trị từ - 2147483648 ... 2147483647*). Các biến này sẽ chứa lần lượt là tổng của các số chẵn có trong dãy, tổng các số lẻ có trong dãy và tổng tất cả các số có trong dãy.

⇨ Đầu tiên bạn sẽ nhập vào số m.

⇨ Kế đến bạn sẽ nhập vào số n. lưu ý số n phải lớn hơn m.

*Ví dụ:* Bạn nhập m=1, n=10

⇨ Vòng lặp for đầu tiên này sẽ kiểm tra xem từ 1 đến 10 có bao nhiêu số chẵn, các số chẵn đó là những số nào, nó chạy từ m đến n, tức là chạy từ 1 đến 10. (Kiểm tra các số từ 1, 2, 3, ....., 9, 10).

⇨ Muốn biết một số là số chẵn, ta kiểm tra số dư của phép chia số đó cho 2, nếu số dư là số 0, ta biết đó là số chẵn.

Ví dụ:

4 chia 2 có số dư là 0, vậy 4 là số chẵn.

7 chia 2 có số dư là 1, vậy 7 không phải là số chẵn.

⇨ Trong C++ có phép toán tìm số dư, đó là phép toán %.

⇨ Phát biểu If ở trong vòng lặp for này sẽ kiểm tra việc này, tức là kiểm tra (i%2) có bằng 0 hay không *(i sẽ chạy lần lượt từ 1 đến 10)*. Nếu thoả, nó sẽ tăng biến k lên 1 và gán số chẵn này vào mảng theo chỉ số k vừa tăng. Cuối cùng khi chạy xong vòng lặp for, các số chẵn đều được gán vào trong các phần tử của mảng có tên Chan, và giá trị của k cho ta biết số số chẵn có trong mảng. Nhờ số k này mà khi cho xuất dữ liệu là các số chẵn ra màn hình, ta chỉ cần cho vòng lặp for chạy từ 1 đến k là có thể xuất ra đầy đủ tránh mất thời gian.

⇨ Sau khi đã nhập xong, để xuất các số chẵn ra màn hình, ta dùng vòng lặp for có i chạy từ 1 đến k, để xuất các phần tử của mảng lần lượt từ Chan[1] đến Chan[k], các số chẵn này được cách nhau một vài khoảng trắng, bạn có thể theo dõi trong câu lệnh ở trên.

⇨ Tương tự như trên, các bạn có thể kiểm tra một số có phải là số lẻ hay không bằng cách lấy số đó Modulo cho 2, nếu bằng 1 (hoặc có thể dùng khác 0 cũng được), thì chắc chắn số này là số lẻ, bạn sẽ gán vào mảng lẻ với chỉ số q, sự việc cũng giải thích tương tự như trong trường hợp số chẵn ở trên.

⇨ Vậy sau khi đã xuất các số chẵn, các số lẻ, bây giờ các bạn phải tính tổng các số chẵn đó.

⇨ Việc tính tổng này rất là dễ dàng, bạn chỉ cần cho vòng lặp for chạy từ 1 đến k, mỗi lần xuất ra một số chẵn, bạn cộng dồn nó vào biến Tong1, biến Tong1 lúc đầu có giá trị bằng 0. Sau khi vòng lặp for chạy xong, bạn sẽ có biến Tong1 là kết quả mà bạn cần tìm.

⇨ Bây giờ bạn chỉ cần xuất nó ra màn hình, sau mỗi lần xuất một dữ kiện nào đó, chúng tôi dùng lệnh cout<<'\n' để xuống dòng cho đẹp.

⇨ Tương tự như vậy, bạn dùng vòng lặp for chạy từ 1 đến q để tính được tổng các số lẻ và cho vào biến Tong2.

⇨ Sau khi đã có giá trị của biến Tong1, Tong2. Bạn chỉ cần cộng hai biến này lại, bạn sẽ được tổng của tất cả các số có trong dãy.

⇨ Trong C++ không cho phép mảng động (*mảng có số phần tử thay đổi*), chỉ số mảng trong C++ phải là hằng số thật sự, không thể là biến số.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Giả sử bạn nhập m=1, và n=10.

⊙ Kết quả trên màn hình sẽ hiện ra như sau:

```
Ban tinh tu so: 1
Den so: 10
Cac so chan tu m den n la:
2  4  6  8  10
Cac so le tu m den n la:
1  3  5  7  9
Tong cac so chan tu m den n la: 30
Tong cac so le tu m den n la: 25
Tong cac so chan va so le tu m den n la: 55
```

⊙ Bây giờ bạn chạy thử chương trình, bạn nhập m=12, n=50. Bạn sẽ có kết quả như sau:

```
Ban tinh tu so: 12
Den so: 50
Cac so chan tu m den n la:
12  14  16  18  20 ..................... 50
Cac so le tu m den n la:
13  15  17  19  21 ..................... 49
Tong cac so chan tu m den n la: 620
Tong cac so le tu m den n la: 589
Tong cac so chan va so le tu m den n la: 1209
```

# ✍ Thông báo dãy 10 số vừa nhập có phải là dãy tăng hay không

Quy ước, dãy tăng là một dãy có số thứ nhất sẽ nhỏ hơn hoặc bằng số thứ hai, ngược lại là không tăng. *(Nếu quy ước dãy tăng là một dãy có số thứ hai lớn hơn số thứ nhất, thì bạn sửa code lại một chút bằng cách bỏ dấu bằng "=").*

**// DayTang.cpp**

```
#include <iostream.h>
#include <conio.h>
int  Day[10];
int  a,b,i,j,n;
main()
{
  // Nhap vao mot day 10 so
   for (i=1; i<=10; i++)
      {
          cout<<"Ban nhap phan tu Day["<<i<<"]= ";
          cin>>Day[i];
      }
  // In ra man hinh Day so vua nhap
    cout<<"Day so ban nhap la:  ";
    for (i=1; i<=10; i++)
         cout<<Day[i]<<"  ";
  // Kiem tra xem day vua nhap co phai la day tang
    for (i=1; i<=9; i++)
        {
            a=Day[i];
            j=i+1;
            b=Day[j];
            if (a<=b)
               n=n+1;
        }
 cout<<'\n';
 // Day co 10 phan tu nen so sanh 9 lan
    if (n==9)
       cout<<"Day ma ban vua nhap la day tang.";
    else
       cout<<"Day ma ban vua nhap, khong phai la day tang.";
 getch ();
 return 0;
}
```

⇨ Khai báo mảng một chiều có tên là Day[10] có kiểu nguyên, dùng để chứa 10 phần tử nhập vào, các phần tử nhập vào là các số nguyên.

⇨ Biến i có kiểu nguyên, dùng làm biến chạy trong các vòng lặp for.

⇨ Biến j cũng có kiểu nguyên, biến này dùng để gán cho một chỉ số của mảng, dùng để lấy phần tử thứ i+1 để so sánh.

⇨ Biến a, b có kiểu nguyên dùng để chứa lần lượt các phần tử của mảng để so sánh.

⇨ Biến n có kiểu nguyên, dùng làm biến đếm số lần số sau lớn hơn số trước. Có 10 số nên có 9 lần so sánh.

So sánh phần tử thứ hai với phần tử thứ nhất của mảng.

So sánh phần tử thứ ba với phần tử thứ hai của mảng.

.....

So sánh phần tử thứ mười với phần tử thứ chín của mảng.

⇨ Vòng lặp for có i chạy từ 1 đến 10 sẽ cho chúng ta nhập vào 10 phần tử của mảng.

⇨ Vòng lặp for thứ hai chạy từ 1 đến 10 để xuất ra 10 phần tử vừa nhập.

⇨ Vòng lặp for thứ ba chạy từ 1 đến 9 để kiểm tra dãy gồm 10 phần tử có phải là dãy tăng hay không. Biến a là phần tử thứ i, biến b là phần tử kế tiếp tức là i+1. Nếu mỗi lần a<=b thì biến đếm n sẽ tăng lên một đơn vị, có 10 phần tử tất cả, vậy phải kiểm tra 9 lần.

⇨ Sau vòng lặp for chạy từ 1 đến 9 kiểm tra xong, nếu biến n=9, tức là không có số nào đứng sau nhỏ hơn số đứng trước, vậy báo ra dãy tăng. Ngược lại nếu n khác 9, nói đúng hơn là n nhỏ hơn 9, thì báo không phải là dãy tăng.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Giả sử bạn nhập lần lượt 10 phần tử là các số nguyên như sau:

Day[1]=12
Day[2]=14
Day[3]=16
Day[4]=18
Day[5]=120
Day[6]=122
Day[7]=124
Day[8]=126
Day[9]=128

Day[10]=128

⊙ Bạn sẽ có thông tin trên màn hình như sau:

Day so ma ban nhap vao la: 12 14 16 18 120 122 124 126 128 128
Day ma ban vua nhap la day tang.

⊙ Bây giờ bạn chạy lại chương trình và nhập vào 10 phần tử như sau:

Day[1]=12
Day[2]=14
Day[3]=16
Day[4]=18
Day[5]=120
Day[6]=12
Day[7]=124
Day[8]=126
Day[9]=128
Day[10]=128

Bạn sẽ có thông tin trên màn hình như sau:

Day so ma ban nhap vao la: 12 14 16 18 120 12 124 126 128 128
Day ma ban vua nhap, khong phai la day tang.

## ✍ In ra n số chẵn và lẻ đầu tiên của một mảng

Giả sử bạn muốn xem 9 số chẵn và 9 số lẻ đầu tiên, bạn sẽ có như sau:

9 số chẵn đầu tiên là:
2 4 6 8 10 12 14 16 18
9 số lẻ đầu tiên là
1 3 5 7 9 11 13 15 17

**// n_sochanledautien.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,n;
int chan[1000];
int le[1000];
int Demchan;
int Demle;
```

```
main()
{
    clrscr();
    cout<<"Ban muon in ra bao nhieu so chan va so le dau tien:  ";
    cin>>n;
    // Nhap vao mang n so chan, n so le dau tien
    Demchan=0;
    Demle=0;
    for (i=1; i<=(n*2); i++)
        if (i%2==0)
          {
              Demchan=Demchan+1;
              chan[Demchan]=i;
          }
        else
          {
              Demle=Demle+1;
              le[Demle]=i;
          }
    // In ra n so chan, n so le dau tien
        cout<<n<<"So chan dau tien la:  ";
        for (j=1; j<=Demchan; j++)
            cout<<chan[j]<<"  ";
        cout<<"\n";
        cout<<n<<"So le dau tien la:  ";
        for (j=1; j<=Demle; j++)
            cout<<le[j]<<"  ";
    getch();
    return 0;
}
```

⇨ Để giải quyết yêu cầu trên, bạn thấy khi ta muốn in ra n số chẵn và n số lẻ đầu tiên, ta chỉ cần cho một vòng for chạy từ 1 đến 2*n . Ta lọc ra các số chẵn cho vào một mảng riêng, số lẻ vào một mảng riêng. Để có kết quả, ta chỉ cần in bằng vòng lặp for.

⇨ Đầu tiên ta nhập số n để xác định n số chẵn, n số lẻ đầu tiên, số n<=1000.

⇨ Trong C++, chúng tôi dùng phép toán Modulo để lọc ra số chẵn và số lẻ. Ứng với số nào, chúng tôi cho vào mảng đó.

⇨ Bài này, không cần dùng biến đếm, vì chúng ta biết chắc chắn số số chẵn và số số lẻ là n *(vì từ 1 đến 2*n sẽ luôn luôn có n số chẵn và n số lẻ)*.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Nhập vào số 30, bạn sẽ có kết quả như sau:

15 số chẵn đầu tiên là:
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
15 số lẻ đầu tiên là:
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

## In ra

- **n số lẻ bình phương đầu tiên**
- **n số chẵn bình phương đầu tiên**
- **Tổng bình phương của n số lẻ đầu tiên**
- **Tổng bình phương của n số chẵn đầu tiên**

```
// Binhphuong.cpp
#include <iostream.h>
#include <conio.h>
int i,j,n;
long int Chan[100];
long int Le[100];
int DemChan;
int DemLe;
long int TongChan;
long int TongLe;
main()
{
    clrscr();
    cout << " Ban can tinh bao nhieu so ";
    cin >> n;
    DemChan=0;
    DemLe=0;
```

```
        TongChan=0;
        TongLe=0;
        for (i=1; i<=(n*2); i++)
            if (i%2==0)
              {
                DemChan=DemChan+1;
                Chan[DemChan]=i;
              }
            else
              {
                DemLe=DemLe+1;
                Le[DemLe]=i;
              }
        cout<<"Binh phuong "<<n<<" so chan dau tien la: ";
        for (j=1; j<=DemChan; j++)
            cout<<Chan[j]<<"^2, ";
        cout<<"\n";
        cout<<"Binh phuong "<<n<<" so le dau tien la:  ";
        for (j=1; j<=DemLe; j++)
            cout<<Le[j]<<"^2, ";
        cout<<"\n";
        for (j=1; j<=DemChan; j++)
            TongChan=TongChan+Chan[j]*Chan[j];
        for (j=1; j<=DemLe; j++)
            TongLe=TongLe+Le[j]*Le[j];
        cout<<"Tong binh phuong "<<n<<" so chan dau tien la:
              "<<TongChan<<"\n";
        cout<<"Tong binh phuong "<<n<<"so Le dau tien la:"<<TongLe<<"\n";
    getch();
    return 0;
}
```

⇨ Đầu tiên là bạn nhập vào số n, với n là số số chẵn và số số lẻ cần tính.

⇨ Vòng lặp for chạy từ 1 đến n*2 cùng với phép toán Modulo để lấy ra n số chẵn và n số lẻ, và được chứa vào mảng của nó.

⊙ Giả sử bạn nhập n=4, bạn sẽ có những thông tin như sau:

Bình phương 4 số chẵn đầu tiên là:

2^2, 4^2, 6^2, 8^2

Bình phương 4 số lẻ đầu tiên là:

1^2, 3^2, 5^2, 7^2

Tổng bình phương của 4 số chẵn đầu tiên là: 120

Tổng bình phương của 4 số lẻ đầu tiên là: 84

## In ra kết quả a x b theo thuật toán sau:

Bạn thiết kế và viết chương trình để nhân 2 số nguyên dương a và b *(a là số thứ nhất với điều kiện a>0, b là số thứ hai, b>=0)* với thuật toán như sau:

Chia đôi số thứ nhất chỉ lấy phần nguyên và gấp đôi số thứ 2. Lặp lại quá trình trên cho đến khi số thứ nhất bằng 1.

Tổng các số thứ 2 tương ứng với các trị lẻ của số thứ nhất sẽ là tích a nhân b cần tính.

Bạn nhập số a, số b.

Chương trình sẽ báo ra kết quả axb.

*Ví dụ:*

Nhân 83 và 20

| | | | | | | |
|---|---|---|---|---|---|---|
| 83 | 41 | 20 | 10 | 5 | 2 | 1 |
| 20 | 40 | 80 | 160 | 320 | 640 | 1280 |

Tích của 83 nhân 20 tính theo giải thuật của đầu bài cho = 1660

(Thử kiểm lại, tương ứng với các vị trí lẻ của số thứ nhất, ta thấy các số thứ hai là: 20+40+320+1280=1660)

```
// Nhancachkhac.cpp
#include <iostream.h>
#include <conio.h>
int i,j,n,m,k,r,Tn,Tm;
int Soa[20];
int Sob[20];
long int Tong;
main()
```

```
{
    clrscr();
    cout<<"Ban nhap so thu nhat n>0:  ";
    cin>>n;
    cout<<"Ban nhap so thu hai m>=0:  ";
    cin>>m;
    i=2;
    Tn=n;
    Tm=m;
    if ((n>0)&&(m>=0))
          // Tim chi so i, i+2 la so cac phan tu khi chia xong
     {
        while ((n/2)!=1)
           {
             n=n/2;
             i=i+1;
           }
        Soa[1]=Tn;
        Sob[1]=Tm;
        n=Tn;
        m=Tm;
        // Neu le thi cong cho Tong
        if (Soa[1]%2==1)
           Tong=Tong+Sob[1];
       // Vong for chay tu phan tu thu 2 den i-1, chua cac so nhan doi
        for (k=2; k<=i; k++)
           {
             Sob[k]=m*2;
             m=Sob[k];
           }
        // Tuong tu chua so chia doi, neu la so le lay so tuong ung cong
            vao tong
        for (j=2; j<=i; j++)
           {
             Soa[j]=n/2;
             n=Soa[j];
```

```
            if (n%2==1)
                Tong=Tong+Sob[j];
        }
      cout<<"Day so thu nhat va thu hai sau khi da nhan va chia"<<"\n";
      for (r=1; r<=i; r++)
        cout<<Soa[r]<<"   ";
      cout<<"\n";
      for (r=1; r<=i; r++)
        cout<<Sob[r]<<"   ";
        cout<<"\n";
        cout<<"Tich cua hai so "<<Tn<<"va " <<Tm<<" theo thuat
                toan tren la:  "<<Tong;
    }
    else
      cout<<"Ban xem lai hai so vua nhap  ";
   getch();
   return 0;
}
```

⇨ Đầu tiên bạn nhập hai số n và m theo điều kiện của đầu bài là n>0, m>=0

⇨ Gán số n và m cho các biến tạm, bạn sẽ thấy công dụng của việc dùng biến tạm này.

⇨ Tiếp đến thực hiện vòng lặp while để tính được số các số khi chia số thứ nhất cho 2 đến khi bằng 1. Theo vòng lặp này biến i ban đầu phải bằng 2 thì mới bằng số phần tử có được

*Ví dụ:* Số thứ nhất bằng 20

Vòng lặp thực hiện cho đến khi nào 20/2<>1

20:2=10  i=1
10:2=5  i=1+1=2
5:2=2  i=2+1=3
2:2=1  Thoát khỏi vòng lặp không thỏa ĐK

Thực ra ta chia số 20 cho 2 được dãy số như sau:

20  10  5  2  1

⇨ Dãy số có 5 số, ở trên i=3 vì mất số 20 và số 1, tức là mất số đầu và số cuối. Nên biến i ban đầu phải bằng 2. Vậy khi chia một số n cho 2

để được một dãy số như thuật toán trên, bạn luôn luôn mất số đầu và số cuối, do đó biến đếm ban đầu phải bằng 2.

⇨ Gán số n cho phần tử thứ nhất của mảng Soa. Gán số m cho phần tử thứ nhất của mảng Sob.

⇨ Gán lại giá trị tạm cho hai biến m và n, phải gán lại vì trong vòng lặp trên giá trị của biến n đã thay đổi.

⇨ Vậy trong hai mảng đã có một phần tử, ta kiểm tra nếu phần tử thứ nhất ở mảng Soa có phải là số lẻ hay không, nếu là số lẻ thì lấy giá trị tương ứng ở mảng Sob cộng vào biến tổng.

⇨ Vòng lặp for chạy từ 2 đến i để nhập các giá trị cho mảng Sob, (các giá trị này là các phần tử đã được nhân đôi).
Vòng lặp for kế tiếp cũng chạy từ 2 đến i để nhập vào mảng Soa (các giá trị đã được chia đôi), mỗi giá trị sẽ được kiểm tra xem có phải là số lẻ hay không, nếu là số lẻ thì lấy giá trị tương ứng ở mảng Sob cộng vào biến Tổng. Sau vòng lặp for này giá trị của biến Tổng sẽ là kết quả của phép nhân hai số.

⇨ Bây giờ ta cho in ra màn hình các yêu cầu của đầu bài, các vòng lặp for tiếp theo và các phát biểu cout và toán tử <<, sẽ thực hiện những điều trên.

⊙ Nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Bạn nhập số thứ nhất n=40, số thứ hai m=50. Bạn sẽ có kết quả như sau:

Dãy số thứ nhất và thứ hai sau khi đã nhân và chia như sau:

| | | | | | |
|---|---|---|---|---|---|
| 40 | 20 | 10 | 5 | 2 | 1 |
| 50 | 100 | 200 | 400 | 800 | 1600 |

Tích của hai số 40 và 50 theo thuật toán trên là 2000

⊙ Kiểm tra lại bạn thấy trong dãy số thứ nhất có số 5 và số 1 là số lẻ, và tương ứng với hai giá trị lẻ này là 400 và 1600, vậy 400+1600=2000.

# In ra dãy Fibonacci

Dãy Fibonacci được xác định như sau:

+ $N_0=N_1=1$
+ $N(n)=N(n-1)+N(n-2)$ $(n>=2)$

Viết chương trình in ra n+1 số Fibonaci với n là số được nhập vào. In ra số Fibonaci thứ Nn.

```
// Fibonaci.cpp
#include <iostream.h>
#include <conio.h>
int n,i;
int So[1000];
main()
{
    cout<<"Ban nhap so nguyen n (Dieu kien n>=2), n=  ";
    cin>>n;
    if (n<2)
        cout<<"Ban nhap so n chua dung theo yeu cau cua dau bai ";
    else
     {
        So[0]=1;
        So[1]=1;
        for (i=2; i<=n; i++)
           So[i]=So[i-1]+So[i-2];
        cout<<"\n";
        cout<<"* n+1 so Fibonaci tu N0 den N "<<n<<" la: ";
        cout<<So[0]<<"  "<<So[1]<<"   ";
        for (i=2; i<=n; i++)
           cout<<So[i]<<"  ";
        cout<<"\n";
        cout<<"So Fibonaci N "<<n<<"la:  "<<So[n];
    }
    getch();
    return 0;
}
```

⇨ Nhập vào số n, nếu bạn nhập n nhỏ hơn 2, chương trình sẽ báo là bạn nhập chưa đúng theo yêu cầu, ngược lại, gán hai giá trị N0 và N1 là 1 cho mảng.

⇨ Vòng lặp for chạy từ 2 đến n để nhập các số Fibonaci cho mảng, các số Fibonaci được xác định qua công thức So[i]=So[i–1]+So[i–2]

⇨ In ra dãy số Fibonaci, và số Fibonaci thứ n

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình

⊙ Bạn nhập n=1. Bạn sẽ được chương trình báo là bạn nhập số n chưa đúng

⊙ Bạn chạy lại chương trình và nhập n=6, bạn sẽ được thông báo như sau:

```
n+1 số  Fibonaci từ N0 đến N6 là:
1    1    2    3    5    8    13
Số Fibonaci N6 là:  13
```

⊙ Chạy lại chương trình và nhập n=12, bạn sẽ được thông báo như sau:

```
N+1 số  Fibonaci từ N0 đến N12 là:
1  1  2  3  5  8  13  21  34  55  89  144  233
Số Fibonaci N12 là:  233
```

# Nhập ma trận 4x4, báo ra

**❑ Số các phần tử lớn hơn 0, liệt kê các phần tử lớn hơn 0 đó, tính tổng của chúng.**

**❑ Số các phần tử nhỏ hơn 0, liệt kê các phần tử nhỏ hơn 0 đó, tính tổng của chúng.**

**❑ Số các phần tử bằng 0.**

```
// Manghaichieu.cpp
#include <iostream.h>
#include <conio.h>
int Mang[5][5];
int i,j;
long int Tong1, Tong2;
int SPT1,SPT2,SPT3;
main()
{
    clrscr();
    cout<<"Ban nhap cac phan tu cho mang: "<<"\n";
    for (i=1; i<=4; i++)
        for (j=1; j<=4; j++)
```

```
        {
            cout<<"Mang["<<i<< "]["<<j<<"]= ";
            cin>>Mang[i][j];
        }
cout<<"Cac phan tu > 0: " ;
for (i=1; i<=4; i++)
   for (j=1; j<=4; j++)
       if (Mang[i][j]>0)
        {
            Tong1=Tong1+Mang[i][j];
            SPT1=SPT1+1;
            cout<<Mang[i][j]<<"   ";
        }
        cout<<"\n";
        cout<<"So phan tu lon hon 0 la:  "<<SPT1<<"\n";
        cout<<"Tong so cac phan tu lon hon 0 la:  "<<Tong1<<"\n";
         cout<<"Cac phan tu < 0: ";
for (i=1; i<=4; i++)
   for (j=1; j<=4; j++)
       if (Mang[i][j]<0)
        {
            Tong2=Tong2+Mang[i][j];
            SPT2=SPT2+1;
            cout<<Mang[i][j]<<"   ";
        }
        cout << "\n";
        cout << " So phan tu nho hon 0 la: " << SPT2 << "\n";
        cout <<"Tong so cac phan tu nho hon 0 la: "<<Tong2 << "\n";
         cout<<"Cac phan tu= 0: " ;
for (i=1; i<=4; i++)
   for (j=1; j<=4; j++)
       if (Mang[i][j]==0)
        {
            SPT3=SPT3+1;
            cout<<Mang[i][j]<<"   ";
        }
```

```
            cout<<"\n";
            cout<<"So phan tu bang 0 la:  " <<SPT3<<"\n";
            cout<<" Tong so cac phan tu bang 0 la: 0  ";
    getch();
    return 0;
}
```

⇨ Trong bài này, chúng ta sử dụng mảng hai chiều, trong C++ bạn sẽ khai báo mảng hai chiều như trên, nó chỉ khác mảng một chiều một chút, đó là thêm một miền. Vậy nếu có một miền thì gọi là mảng một chiều, có hai miền là mảng hai chiều.

⇨ Nếu bạn khai báo Mang[4][4] và dùng hai vòng lặp for lồng nhau như trên để nhập, các bạn chỉ nhập được 12 phần tử chứ không phải là 16 (vì thành phần cuối của mảng là phần tử trống). Vì vậy chúng tôi phải khai báo là Mang[5][5].

⇨ Đầu tiên là bạn nhập các phần tử vào ma trận 4x4. Bạn dùng hai vòng for lồng nhau (vì đây là mảng hai chiều).

⇨ Sau đó bạn sẽ xuất ra các phần tử lớn hơn 0, số phần tử lớn hơn 0, tổng số các phần tử lớn hơn 0, qua hai vòng lặp for lồng nhau cùng với phát biểu If kiểm tra điều kiện.

⇨ Tương tự xuất ra các phần tử nhỏ hơn 0, bằng 0.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Giả sử bạn nhập các phần tử đều bằng 1, bạn sẽ có kết quả thông báo như sau:

```
Các phần tử lớn hơn 0 là:
1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
Số các phần tử lớn hơn 0 là: 16
Tổng số các phần tử lớn hơn 0 là: 16
Các phần tử nhỏ hơn 0 là:
Số các phần tử nhỏ hơn 0 la: 0
Tổng số các phần tử nhỏ hơn 0 la : 0
Các phần tử bằng 0 là: 0
Số các phần tử bằng 0 là: 0
Tổng các phần tử bằng 0 la : 0
```

# ✍ Nhập vào 11 số, kiểm tra có phải là một cấp số cộng hay không

Trong chương trình trên, các bạn đã kiểm tra ba số xếp theo một thứ tự nào đó có phải là cấp số cộng hoặc cấp số nhân hay không.

Bài trên, chúng tôi muốn giới thiệu cho các bạn về hai định lý cấp số cộng và cấp số nhân áp dụng với ba số phân biệt.

Chương trình này giới thiệu đến các bạn dãy số dài hơn, gồm 11 số

Bạn sẽ viết chương trình để bắt đầu nhập vào 11 số, sau đó kiểm tra dãy 11 số vừa nhập vào theo thứ tự có phải là một cấp số cộng hay không. Bạn sẽ kiểm tra bằng cách so sánh số sau có bằng với số trước cộng với công sai hay không.

Nếu là cấp số cộng, bạn tính tổng Sn của dãy số đó.

```
// Capsocong2.cpp
#include <iostream.h>
#include <conio.h>
int i,j,n;
float a,b, Congsai;
float CSC[12];
main()
{
    clrscr();
    for (i =1;i<=11;i++)
        {
            cout << " Ban nhap vao phan tu thu CSC["<< i<< "] =   ";
            cin >> CSC[i];
        }
    Congsai=CSC[2]-CSC[1];
    cout<<"Day so: ";
    for (i=1; i<=11; i++)
        cout<<CSC[i]<<"  ";
    if (Congsai == 0)
        cout<<"\n Day so tren khong phai la cap so cong, vi cong sai= 0.";
    else
```

```
        for (i=1; i<=10; i++)
          {
            a=CSC[i];
            j=i+1;
            b=CSC[j];
            if (b==a+Congsai)
              n=n+1;
          }
      if (n==10)
        {
          if (Congsai>0)
            {
              cout<<"\n Day so ma ban nhap vao o tren la mot
                    cap so cong tien, voi cong sai la "<<Congsai;
              cout<<"\n Tong Sn cua cap so cong tren la= "<<
                          11*(CSC[1]+CSC[11])/2;
            }
          if (Congsai<0)
            {
              cout<<"\n Day so ma ban nhap vao o tren la mot
                    cap so cong thoai, voi cong sai la"<<Congsai;
              cout<<"\n Tong Sn cua cap so cong tren la= "<<
                          11*(CSC[1]+CSC[11])/2;
            }
        }
      else
          cout<<"\n Day so ma ban nhap vao o tren khong phai la
                    mot cap so cong. ";
  getch();
  return 0;
}
```

⇨ Trong bài này chúng tôi dùng biến i có kiểu nguyên để làm biến chạy trong các vòng lặp for. Biến j có kiểu nguyên dùng làm chỉ số các phần tử trong mảng để so sánh. Biến n có kiểu nguyên, dùng để đếm khi điều kiện số sau = số trước+Cong sai, nhờ biến này, chúng ta sẽ biết dãy số nhập vào có phải là một cấp số cộng hay không.

⇨ Biến a, b có kiểu là số thực dùng để chứa giá trị lần lượt của các phần tử trong mảng để so sánh.

⇨ Biến Congsai có kiểu là số thực, dùng để chứa giá trị công sai của cấp số cộng.

⇨ Khai báo mảng một chiều có tên CSC[12], có kiểu số thực.

⇨ Vòng lặp for đầu tiên dùng để nhập vào 11 phần tử.

⇨ Cho xuất hiện 11 phần tử trên màn hình, mỗi phần tử cách nhau bằng 2 khoảng trắng.

⇨ Tính công sai

⇨ Kiểm tra điều kiện của công sai, nếu bằng 0, báo ngay câu "không phải là cấp số cộng", ngược lại sẽ kiểm tra tiếp (*vì trong trường hợp ngược lại này, vẫn có hai trường hợp xảy ra, có hoặc không*).

⇨ Vòng lặp For chạy từ 1 đến 10 để kiểm tra 10 lần, kiểm tra phần tử thứ hai với phần tử thứ nhất, kiểm tra phần tử thứ ba với phần tử thứ hai ... kiểm tra phần tử thứ mười một với phần tử thứ mười. (Phần code này chắc bạn còn nhớ, trong bài kiểm tra một dãy có phải là dãy tăng hay không, bạn thấy nó sẽ được ứng dụng trong rất nhiều trường hợp, bạn cần nhớ).

⇨ Nếu biến n=10, thì chắc chắn dãy số trên là một cấp số cộng, ngược lại thì không phải là một cấp số cộng.

⇨ Bạn có thể tính tổng Sn của một cấp số cộng bằng cách dùng vòng For chạy từ phần tử đầu đến phần tử cuối có trong mảng, dùng biến tổng để cộng dồn để có kết quả cuối cùng, tuy nhiên trong bài này chúng tôi dùng công thức:

Sn=n*(U1+Un)/2

⊙ Bạn nhấn phím Ctrl+F9 để chạy thử chương trình.

⊙ Bạn nhập vào 11 phần tử như sau:

```
CSC[1]=1
CSC[2]=2
CSC[3]=3
CSC[4]=4
CSC[5]=5
CSC[6]=6
```

CSC[7]=7
CSC[8]=8
CSC[9]=9
CSC[10]=10
CSC[11]=11

Bạn sẽ có kết quả

Day so la: 1 2 3 4 5 6 7 8 9 10 11
Day so ma ban nhap o tren la mot cap so cong tien, voi cong sai la: 1
Tong Sn cua cap so cong tren la: 66

⊙ Chạy lại chương trình, và nhập vào 11 phần tử như sau

CSC[1]=2
CSC[2]=4
CSC[3]=6
CSC[4]=8
CSC[5]=10
CSC[6]=12
CSC[7]=14
CSC[8]=16
CSC[9]=18
CSC[10]=20
CSC[11]=22

+ Bạn sẽ có kết quả

Day so la: 2 4 6 8 10 12 14 16 18 20 22
Day so ma ban nhap o tren la mot cap so cong tien, voi cong sai la: 2
Tong Sn cua cap so cong tren la: 132

## ✍ Nhập vào 11 số, kiểm tra có phải là một cấp số nhân hay không

**// Capsonhan2.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,n;
float a,b, Congboi;
float Sn;
float CSN[12];
```

```
main()
{
    clrscr();
    for (i=1; i<=11; i++)
        {
          cout<<"Ban nhap vao phan tu thu CSN["<< i<< "]=  ";
          cin>>CSN[i];
        }
         cout<<"Day so: ";
    for (i=1; i<=11; i++)
        cout<<CSN[i]<<"  ";
         Congboi=(CSN[2]/CSN[1]);
    if ((Congboi==0)||(Congboi==1))
         cout<<"\nDay so ban vua nhap o tren khong phai la cap so nhan";
    else
         for (i=1; i<=10; i++)
           {
              a=CSN[i];
              j=i+1;
              b=CSN[j];
              if (a*Congboi==b)
                   n = n+1;
           }
         if (n == 10)
            {
                if ((Congboi>1)&&(CSN[1]>0))
                   {
                     cout<<"\n Day so tren la cap so nhan tien voi cong
                                   boi la: "<<Congboi;
                     Sn=0;
                     for (i=1; i<=11; i++)
                          Sn=Sn+CSN[i];
                     cout<<"\nTong sn cua cap so nhan tren la:"<<Sn;
                   }
                if ((Congboi>0)&&(CSN[1]>0)&&(Congboi<1))
                   {
```

```
                cout<<"\n Day so tren la cap so nhan thoai voi cong
                        boi la: "<<Congboi;
                Sn=0;
                for (i=1; i<=11; i++)
                    Sn=Sn+CSN[i];
                cout<<"\n Tong sn cua cap so nhan tren la: "<<Sn;
              }
            if (Congboi<0)
              {
                cout<<"\n Day so tren la cap so nhan co cac phan
                  tu lan luot thay doi dau, voi cng boi la "<<Congboi;
                Sn=0;
                for (i=1; i<=11; i++)
                    Sn=Sn+CSN[i];
                cout<<"\n Tong sn cua cap so nhan tren la: "<<Sn;
              }
        }
      else
        cout<<"\n Day so tren khong phai la mot cap so nhan. ";
  getch ();
  return 0;
}
```

⇨ Bài này cũng có nét giống như trong chương trình cấp số cộng ở trên, việc khai báo các biến, kiểu của chúng, cách thể hiện và chức năng của nó cũng tương tự.

⇨ Có thêm biến Sn có kiểu thực, biến này sẽ chứa tổng của 11 số hạng của cấp số nhân. Trong phần này, chúng tôi không dùng công thức, chúng tôi dùng vòng lặp For.

⇨ Các biểu thức điều kiện theo cách giải quyết của cấp số nhân, bạn đối chiếu và so sánh với các điều kiện của cấp số cộng, để rút ra cách nhớ cho mình.

⊙ Chạy thử chương trình trên và nhập vào 11 phần tử như sau, bạn sẽ có kết quả:

```
Ban nhap vao phan tu CSN[1]=1
Ban nhap vao phan tu CSN[2]=2
```

```
Ban nhap vao phan tu CSN[3]=4
Ban nhap vao phan tu CSN[4]=8
Ban nhap vao phan tu CSN[5]=16
Ban nhap vao phan tu CSN[6]=32
Ban nhap vao phan tu CSN[7]=64
Ban nhap vao phan tu CSN[8]=128
Ban nhap vao phan tu CSN[9]=256
Ban nhap vao phan tu CSN[10]=512
Ban nhap vao phan tu CSN[11]=1024
Day so: 1 2 4 8 16 32 64 128 256 512 1024
Day so tren la cap so nhan tien voi cong boi la 2
Tong Sn cua cap so nhan tren la: 2.047
```

# Chèn một số vào dãy số nguyên tăng dần

Trong bài này, chúng tôi giới thiệu với bạn một thuật toán để chèn một số nguyên vào đúng vị trí thích hợp của một dãy số nguyên tăng dần cho trước.

Bạn thiết kế một chương trình để người sử dụng nhập vào một dãy số tăng dần trong các TextBox, sau đó người sử dụng nhập vào một số nguyên bất kỳ, chương trình phải sắp xếp sao cho số đó chèn vào nơi thích hợp.

*Ví dụ:*

Giả sử dãy số đó có 8 phần tử theo thứ tự như sau:

1 3 6 34 45 56 67 678

Bây giờ nếu như bạn nhập vào số 42, chương trình phải xếp cho bạn như sau:

1 3 6 34 42 45 56 67 678

**// Chenso.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,x,n;
int a[10];
main()
{
```

```
    clrscr();
    cout<<"Ban nhap vao 8 phan tu cua mang ";
    n=9;
    for (i=1; i<=n-1; i++)
        {
            cout<<"Ban nhap phan tu a["<<i<<"]= ";
            cin>>a[i];
        }
    cout<<"Ban nhap vao phan tu can chen: x=  ";
    cin>>x;
    i=1;
    while ((i!=9)&&(x>a[i]))
        i=i+1;
    for (j=0; j<=n-i; j++)
        a[n-j+1]=a[n-j];
        a[i]=x;
    cout<<"Cac phan tu cua mang sau khi chen: ";
    for (j=1; j<=n; j++)
        cout<<a[j]<<"   ";
    getch();
    return 0;
}
```

- Bạn nhấn Ctrl+F9 để chạy thử chương trình
- Bạn nhập lần lượt 8 phần tử của mảng như sau:

  A[1]=1
  A[2]=2
  A[3]=3
  A[4]=4
  A[5]=5
  A[6]=6
  A[7]=7
  A[8]=8

Sau đó giả sử nhập x=5, giả sử bạn nhập số 5, bạn sẽ có thông báo dãy mới như sau:

1 2 3 4 5 5 6 7 8

Bạn cũng nhập 8 phần tử như trên, nhưng nhập giá trị cần chèn là 0, bạn sẽ được thông báo dãy như sau:

0 1 2 3 4 5 6 7 8

Bạn cũng nhập 8 phần tử như trên, nhưng nhập giá trị cần chèn là 9, bạn sẽ được thông báo dãy như sau:

1 2 3 4 5 6 7 8 9

# Tính hiệu hai số nhị phân nguyên dương có chiều dài bằng nhau

⇨ Trong chương trình này các bạn lại tìm hiểu thuật toán tìm hiệu của hai số nhị phân nguyên dương a và b có độ dài bằng nhau với điều kiện a>b.

*Ví dụ:*

Số nhị phân a là 1 1 1 1

Số nhị phân b là 0 1 1 1

Kết quả là 1 0 0 0

Phép trừ hai số nhị phân như sau

0-0=0

0-1=1 (mượn 1)

1-0=1

1-1=0

Thuật toán tính hiệu hai số nhị phân

$a=(a_{n-1}\ a_{n-2} \ldots\ldots a_1\ a_0)_2$

$b=(b_{n-1}\ b_{n-2} \ldots\ldots b_1\ b_0)_2$

Khai báo các mảng có tên a, b và Ketqua để chứa các kí số của số nhị phân a, số nhị phân b và kết quả của số nhị phân a trừ cho số nhị phân b.

```
c=0;
for (j=1; j<=n; j++)
   {
      if (a[j]>=b[j]+c)
       {
         Kq[j]=a[j]-b[j]-c;
         c=0;
```

```
            }
            else
            {
                Kq[i]=a[i]+2-b[i]-c;
                c=1;
            }
        }

// Hieuhaiso_Nhiphan.cpp
#include <iostream.h>
#include <conio.h>
#include <string.h>
int soa[20];
int sob[20];
int i,j,k,m,n,c,x;
int a[20];
int b[20];
int Kq[20];
main()
{
    clrscr();
    cout<<"Ban nhap so nhi phan Soa,Sob. Dieu kien Soa>Sob ";
    cout<<"Ban nhap chieu dai cua hai so nhi phan:  ";
    cin>>x;
    cout<<"So nhi phan soa:";
    for (i=1; i<=x; i++)
        cin>>soa[i];
    cout<<"So nhi phan sob:";
    for (i=1; i<=x; i++)
         cin>>sob[i];
        n=x;
        m=x;
    for (i=1; i<=n; i++)
        a[i]=(soa[n-i+1]);
    for (i=1; i<=m; i++)
        b[i]=(sob[m-i+1]);
```

```
    c=0;
    for (j=1; j<=n; j++)
      {
          if (a[j]>=b[j]+c)
            {
              Kq[j]=a[j]-b[j]-c;
              c=0;
            }
              else
                {
                  Kq[j]=a[j]+2-b[j]-c;
                  c=1;
                }
          }
        cout<<"Chuoi nhi phan so a la: ";
        for(k=n; k>=1; k--)
            cout<<a[k]<<"   ";
            cout<<"\n";
            cout<<"Chuoi nhi phan so b la: ";
        for(k=n; k>=1; k--)
            cout << b[k] << "   ";
            cout << "\n";
            cout<<"Chuoi nhi phan ket qua la: ";
    for(k=n; k>=1; k --)
          cout<<Kq[k]<<"   ";
  getch();
  return 0;
}
```

⇨ Khai báo hai mảng một chiều có kiểu là số nguyên với tên là soa và sob dùng để chứa các kí tự số của hai số nhị phân. Chúng tôi dùng mảng như vậy để thuận tiện khi viết chương trình, bạn cần lưu ý là khi nhập một kí tự xong, bạn phải nhấn phím Enter để tiếp tục nhập kí tự tiếp theo, và cứ như vậy cho đến khi nhập xong các kí tự của số nhị phân.

*Ví dụ:* Bạn nhập một số nhị phân là 1011. Bạn nhập số 1, sau đó nhấn Enter, tiếp tục nhập số 0, nhấn Enter... tiếp tục như vậy cho đến khi nhập xong.

⇨ Bạn khai báo các biến có kiểu nguyên dùng để thực hiện các công việc trong chương trình.

⇨ Bạn khai báo ba mảng một chiều có kiểu nguyên, hai mảng dùng để chứa các kí tự số theo chiều ngược lại, việc này dùng để thuận tiện cho công việc tính toán. Mảng Kq dùng để chứa kết quả của số nhị phân thứ nhất trừ cho số nhị phân thứ hai.

⇨ Chương trình đầu tiên là một câu ghi chú để bạn nhập hai số nhị phân cho đúng.

⇨ Tiếp đến sẽ nhập chiều dài của hai số nhị phân.

⇨ Tiến hành nhập số nhị phân thứ nhất, số nhị phân thứ hai qua hai vòng lặp For.

⇨ Tiến hành gán vào hai mảng a và b các kí tự số của hai mảng soa, sob theo chiều ngược lại. Điều này có nghĩa là phần tử a[1] sẽ chứa phần tử cuối cùng của mảng soa, phần tử a[2] sẽ chứa phần tử áp cuối của mảng soa... tương tự như vậy với mảng b, sob. Điều này để thuận tiện cho việc tính toán như chúng tôi đã đề cập ở trên.

⇨ Tiến hành tính toán và gán vào mảng có tên Kq.

⇨ Báo kết quả ra màn hình số nhị phân thứ nhất, số nhị phân thứ hai, kết quả của số nhị phân thứ nhất trừ cho số nhị phân thứ hai.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình, giả sử bạn nhập chiều dài của hai số nhị phân là 4. Và giá trị của hai số nhị phân là:soa=1111, sob=1000. Bạn sẽ có kết quả như sau:

Chuỗi số nhị phân soa là : 1 1 1 1
Chuỗi số nhị phân sob là : 1 0 0 0
Chuỗi số nhị phân kết quả là : 0 1 1 1

# Tính hiệu hai số thập phân nguyên dương có chiều dài bằng nhau

⇨ Yêu cầu giống như bài trên, nhưng bây giờ là số thập phân. Người sử dụng nhập vào số thập phần a và số thập phân b, với điều kiện là a>b

và độ dài hai số phải bằng nhau. Chương trình sẽ thông báo giống như dạng của bài trên.

***// Hieuhaiso_Thapphan.cpp***

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
int soa[12];
int sob[12];
int i,j,k,m,n,c,x;
int a[12];
int b[12];
int Kq[12];
main()
{
   clrscr();
   cout<<"Ban nhap so thap phan soa,sob. Dieu kien soa>sob ";
   cout<<"Ban nhap chieu dai cua hai so thap phan:  ";
   cin>>x;
   cout<<"So thap phan soa:";
   for (i=1; i <=x; i++)
      cin >> soa[i] ;
   cout << " So thap phan sob:";
   for (i=1; i<=x; i++)
       cin>>sob[i];
      n=x;
      m=x;
   for (i=1; i<=n; i++)
      a[i]=(soa[n-i+1]);
   for (i=1; i<=m; i++)
      b[i]=(sob[m-i+1]);
   c=0;
   for (j=1; j<=n; j++)
      {
          if (a[j]>=b[j]+c)
            {
               Kq[j]=a[j]-b[j]-c;
```

```
            c=0;
        }
          else
              {
                Kq[j]=a[j]+10-b[j]-c;
                c=1;
              }
        }
      cout<<"Chuoi thap phan so a la: ";
      for(k=n; k>=1; k--)
          cout<<a[k]<<"   ";
           cout<<"\n";
           cout<<"Chuoi thap phan so b la: ";
    for(k=n; k>=1; k--)
         cout<<b[k]<<"   ";
          cout<<"\n";
          cout<<"Chuoi thap phan ket qua la: ";
    for(k=n; k>=1; k --)
         cout<<Kq[k]<<"   ";
  getch();
  return 0;
}
```

⇨ Phần giải thích của bài này cũng giống như trên, chỉ khác là hệ thập phân, thay số 2 thành số 10 và các phần ghi chú.

◎ Bạn chạy thử chương trình và nhập chuỗi soa=15, sob=08. Bạn sẽ có kết quả như sau:

Chuỗi số thập phân soa là:15
Chuỗi số thập phân sob là:08
Chuỗi số thập phân kết quả là:07

◎ Bài này bất tiện vì có số 0 đứng trước, không đúng với yêu cầu trong toán học, bạn hãy suy nghĩ và khắc phục xem sao.

# Tính tích Descartes

⇨ Trong chương trình này, chúng tôi muốn giới thiệu đến các bạn cách tính tích Descartes trong máy tính.

⇨ Bạn biết rằng **tích Descartes** của các tập $A_1, A_2, \ldots, A_n$ được kí hiệu bởi $A_1xA_2x\ldots\ldots xA_n$ là tập hợp của các dãy sắp thứ tự $(a_1, a_2, \ldots, a_n)$ trong đó $a_i \in A_i$ với i=1, 2, ... , n . Nói một cách khác

$A_1xA_2x\ldots xA_n = \{(a_1,a_2,\ldots,a_n) \backslash a_i \in A_i$ với i=1, 2, ... , n}

*Ví dụ:*

Tính tích Descartes của AxBxC với .

A={0, 1}

B={1, 2}

C={0, 1, 2}

Tích Descartes AxBxC gồm tất cả các dãy ba phần tử sắp thứ tự (a, b, c) với a∈A, b∈B và c∈C. Từ đó

AxBxC={(0, 1, 0), (0, 1, 1), (0, 1, 2), (0, 2, 0), 0, 2, 1), (0, 2, 2),
(1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)}

⇨ Nếu tính tích Descartes trên giấy thì rất là phức tạp nếu như có nhiều tập hợp, và càng phức tạp hơn nếu như các tập hợp lại có nhiều phần tử. Trong máy tính thì việc này rất dễ dàng, vì nó tính rất nhanh, bạn chỉ cần viết đúng thuật toán.

⇨ Bây giờ bạn thử viết chương trình để báo ra được kết quả như ví dụ trên, người sử dụng nhập vào các phần tử của ba tập hợp A, B và C. Tập hợp A có hai phần tử, tập hợp B có hai phần tử và tập hợp C có ba phần tử. Khi đã nhập xong, người sử dụng click vào nút để xem kết quả, chương trình sẽ báo ra kết quả của tích Descartes AxBxC có dạng như ví dụ trên.

⇨ Khi đã thiết lập xong, các bạn có thể thiết kế tính tích Descartes của nhiều tập hợp hơn, trong tập hợp lại có nhiều phần tử.

⇨ Nhìn vào kết quả của ví dụ trên, phân tích bạn sẽ thấy, đầu tiên với phần tử thứ nhất của tập A, nhóm với phần tử thứ nhất của tập B, phần tử thứ nhất của tập C, ta được một phần tử của AxBxC. Giữ nguyên phần tử của tập A, tập B, lấy phần tử thứ hai của tập C, ta được phần tử thứ hai của AxBxC, vẫn như vậy nhưng lấy phần tử thứ ba của tập C, ta được phần tử thứ ba của AxBxC.

⇨ Tập C hết phần tử, tăng tập B lên 1, chạy lại lấy từ phần tử đầu của tập C cho đến phần tử thứ ba của C.

⇨ Khi các phần tử của tập B hết, ta qua phần tử kế tiếp của tập A, quá trình thực hiện lại như trên cho đến khi thực hiện xong phần tử cuối cùng của tập A.

⇨ Vậy muốn lấy kết quả trên, các bạn phải dùng ba vòng For lồng nhau.

⇨ Với suy luận trên, bạn thấy có bao nhiêu tập hợp, thì sẽ có bấy nhiêu vòng For lồng nhau. Vì kết quả là dãy sắp thứ tự, nên khi đầu bài cho các tập hợp được tính tích Descartes theo thứ tự như thế nào, thì kết quả của dãy cũng phải xếp theo thứ tự đó, chúng tôi nói điều này, để các bạn sắp xếp các vòng lặp for cho đúng với vị trí như yêu cầu của đầu bài.

**// Descartes.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,k;
int a[3];
int b[3];
int c[4];
main()
{
   clrscr();
   cout<<"Ban nhap vao gia tri cua mang a  ";
   for (i=1; i<=2; i++)
      {
          cout<<"a["<<i<<"]= ";
          cin >>a[i];
      }
   cout<<"Ban nhap vao gia tri cua mang b  ";
   for (i=1; i<=2; i++)
      {
          cout<<"b[" <<i<<"]= ";
          cin>>b[i];
      }
          cout<<"Ban nhap vao gia tri cua mang c  ";
   for (i=1; i<=3; i++)
      {
```

```
        cout<<"c[" <<i<<"]= ";
        cin>>c[i];
    }
        cout<<"Tich Descartes cua axbxc la:";
  cout<<"{";
  for (i=1; i<=2; i++)
     for (j=1; j<=2; j++)
          for (k=1; k<=3; k++)
             cout<<"("<<a[i]<<","<<b[j]<<","<<c[k]<<")"<<","<<" ";
             cout<<"}";
  getch();
  return 0;
}
```

⇨ Chúng tôi khai báo ba mảng một chiều như trên.

⇨ Bạn nhập vào các phần tử của ba mảng trên, sau đó dùng ba vòng For lồng nhau để tính tích Descartes của ba tập hợp trên.

◎ Bạn tự chạy thử và nhập vào các phần tử bất kỳ của ba tập hợp, các bạn sẽ thấy kết quả.

# ✍ Tìm kiếm tuyến tính trong một dãy số nguyên phân biệt

Trong bài này, chúng tôi giới thiệu đến các bạn một thuật toán *tìm kiếm tuyến tính* hay *tìm kiếm tuần tự một phần tử* trong một *dãy số nguyên phân biệt*. Thuật toán sẽ như sau:

Nó bắt đầu bằng việc so sánh x (là số cần tìm) với $a_1$. Nếu $x=a_1$, nghiệm là vị trí của $a_1$, tức là 1, nếu $x<>a_1$, so sánh x với $a_2$. Nếu $x=a_2$ nghiệm là vị trí của $a_2$, tức là 2. Nếu $x<>a_2$ so sánh x với $a_3$. Tiếp tục quá trình này bằng cách so sánh x với mỗi số hạng của dãy, cho tới khi tìm được số hạng bằng x, khi đó nghiệm là vị trí của số hạng đó. Nếu toàn dãy đã được kiểm tra mà không xác định được vị trí của x, thì x=0.

**// Timkiem_Tuyentinh.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,n,x;
```

```
int a[20];
main()
{
    clrscr();
    n=18;
    cout<<" Ban nhap vao 18 phan tu cua mang ";
    for (i=1; i<=n; i++)
        {
            cout<<"a["<<i<<"]= ";
            cin>>a[i];
        }
    cout<<"Ban nhap phan tu can tim, x=  ";
    cin>>x;
    i=1 ;
    while ((i<=19)&&(x!=a[i]))
        i=i+1;
        if (i<=n)
            cout<<"Phan tu "<<x<<" ma ban can tim nam o vi tri thu "
                    <<i<<"cua mang";
        else
            cout<<"Khong co phan tu " <<x<<" trong mang ";
        getch();
        return 0;
}
```

⇨ Trong bài trên có khuyết điểm là chỉ tìm được phần tử nào đó ở vị trí đầu tiên, khi gặp thì nó báo vị trí tìm được trong mảng, nó không biết sau đó còn có phần tử nào giống x nữa không.

⇨ Phần code chắc không có gì phức tạp, các bạn có thể tự tìm hiểu.

◎ Chạy thử chương trình, nhập vào 18 phần tử của mảng như sau

a[1]=1
a[2]=1
a[3]=1
a[4]=3
a[5]=1
a[6]=1

a[7]=1
a[8]=1
a[9]=1
a[10]=1
a[11]=1
a[12]=1
a[13]=1
a[14]=1
a[15]=1
a[16]=1
a[17]=1
a[18]=1

➔ Giả sử sau đó bạn nhập phần tử cần tìm là 1.

➔ Bạn được thông báo số 1 mà bạn cần tìm nằm ở vị trí thứ 1.

➔ Bạn thấy trong mảng có rất nhiều phần tử bằng 1.

➔ Để báo được tất cả các vị trí, bạn tham khảo code sau:

```
// Timkiem_nhieuvitri.cpp
#include <iostream.h>
#include <conio.h>
int i,n,x,dem;
int a[19];
int b[19];
main()
{
    clrscr();
    n=18;
    cout<<"Ban nhap vao 18 phan tu cua mang ";
    for (i=1; i<=n; i++)
        {
            cout<<"a["<<i<<"]= ";
            cin>>a[i];
        }
    cout<<"Ban nhap phan tu can tim, x= ";
    cin>>x;
    dem=0;
    for (i=1; i<=18; i++)
```

```
        if (x==a[i])
          {
              dem=dem+1;
              b[dem]=i;
          }
        if (dem>0)
          {
              cout<<"Trong mang co phan tu ma ban can tim. ";
              cout<<"Phan tu "<<x<<" o cac vi tri: ";
              for (i=1; i<=dem; i++)
                  cout<<b[i]<<","<<" ";
          }
        else
          {
              cout<<"\n";
              cout<<"Trong mang khong co phan tu "<<x;
          }
        getch();
        return 0;
}
```

⇨ Mảng a dùng để chứa 18 phần tử nhập vào.

⇨ Mảng b dùng để chứa các vị trí của phần tử khi nó bằng x (*với x là số cần tìm*). Ta dùng biến dem là số lần phần tử trong mảng bằng x. Mỗi lần có phần tử trong mảng bằng x, ta ghi nhận vị trí thứ i này vào mảng b theo chỉ số của biến dem (*biến dem sẽ từ 1 đến số lần bằng x*). Nếu không có phần tử nào bằng x, biến dem sẽ bằng 0.

# ✍ Tìm kiếm nhị phân trong một dãy số nguyên tăng dần

☞ Trong bài trước chúng tôi đã giới thiệu tới bạn thuật toán tìm kiếm tuyến tính. Trong bài này, chúng tôi giới thiệu tới các bạn *thuật toán tìm kiếm nhị phân.*

*Thuật toán này có thể được dùng khi dãy được sắp xếp theo thứ tự tăng dần (Nếu là số thì chúng được sắp từ số nhỏ nhất đến số lớn nhất, nếu là từ thì chúng được sắp theo thứ tự bảng chữ cái).* Thuật toán này được tiến hành bằng cách so sánh phần tử cần xác định vị trí với số hạng ở giữa dãy liệt kê. Sau đó dãy này được tách làm hai dãy con nhỏ hơn có kích thước như nhau, hoặc một trong hai dãy con ít hơn dãy con kia một số hạng. Sự tìm kiếm tiếp tục bằng cách hạn chế tìm kiếm ở một dãy con thích hợp dựa trên việc so sánh phần tử cần xác định vị trí với số hạng giữa dãy liệt kê.

*Ví dụ:* Để tìm số 16 trong dãy liệt kê

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

⇨ Ta tách dãy liệt kê gồm 18 số hạng này thành hai dãy liệt kê nhỏ hơn, mỗi dãy có 9 số hạng, cụ thể là:

1, 2, 3, 4, 5, 6, 7, 8, 9 và

10, 11, 12, 13, 14, 15, 16, 17, 18

⇨ Sau đó ta so sánh số 16 với số hạng cuối cùng của dãy con thứ nhất, vì 9<16, việc tìm kiếm 16 chỉ giới hạn trong dãy liệt kê con thứ hai, từ số hạng thứ 10 đến thứ 18 trong dãy liệt kê ban đầu. Tiếp theo, ta lại tách dãy con gồm 9 số hạng này làm hai dãy con, dãy một gồm 4 phần tử, dãy hai gồm 5 phần tử cụ thể là:

10, 11, 12, 13 và 14, 15, 16, 17, 18

⇨ Kế đến ta so sánh số 16 với số hạng cuối cùng của dãy con thứ nhất, vì 13<16, việc tìm kiếm 16 chỉ giới hạn trong dãy liệt kê con thứ hai, từ số hạng thứ 14 đến thứ 18 trong dãy liệt kê ban đầu. Tiếp theo, ta lại tách dãy con gồm 5 số hạng này làm hai dãy con, dãy một gồm 2 phần tử, dãy hai gồm 3 phần tử cụ thể là:

14, 15 và 16, 17, 18

⇨ Tiếp theo ta so sánh số 16 với số hạng cuối cùng của dãy con thứ nhất, vì 15<16, việc tìm kiếm 16 chỉ giới hạn trong dãy liệt kê con thứ hai, từ số hạng thứ 16 đến thứ 18 trong dãy liệt kê ban đầu. Tiếp theo, ta lại tách dãy con gồm 3 số hạng này làm hai dãy con, dãy một gồm 1 phần tử, dãy hai gồm 2 phần tử, cụ thể là:

16 và 17, 18

⇨ Vì 16 không lớn hơn số hạng lớn nhất của bảng con thứ nhất (cũng là 16), nên việc tìm kiếm giới hạn chỉ ở dãy con thứ nhất chỉ có số 16. Và số 16 đã được tìm ra, nó ở vị trí thứ 16 của dãy liệt kê ban đầu.

**// TimkiemNhiphan.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,m,n,x;
int a[19];
main()
{
    clrscr();
    n=18;
    cout<<"Ban nhap vao 18 phan tu cua mang ";
    for (i=1; i<=n; i++)
        {
            cout<<"a["<<i<<"]= ";
            cin>>a[i];
        }
    cout<<"Ban nhap phan tu can tim, x=  ";
    cin>>x;
    i=1;
    j=n;
    while (i<j)
        {
          m=(i+j)/2;
          if (x>a[m])
              i=m+1;
          else
              j=m;
        }
        if (x==a[i])

          cout<<"Phan tu "<<x<<" ma ban can tim o vi tri: "<<i<<
                "cua mang";
        else
          cout<<"Trong mang khong co phan tu "<<x;
```

```
    getch();
    return 0;
}
```

◎ Chạy thử chương trình, nhập vào 18 phần tử cho mảng a[i] như sau:

1 3 5 6 7 8 9 12 13 45 56 78 88 89 123 134 234 456.

Sau đó bạn nhập số x cần tìm là 78.

Bạn sẽ thấy thông báo, số 78 mà bạn cần tìm ở vị trí thứ 12.

## Trong mảng, tính

- **Số số 0 có trong mảng.**
- **Tổng các số nhỏ hơn 0.**
- **Tổng các số lớn hơn 0.**
- **Tổng các số có trong mảng.**
- **Tổng bình phương của các số nhỏ hơn 0.**
- **Tổng bình phương của các số lớn hơn 0.**
- **Tổng lập phương của các số nhỏ hơn 0.**
- **Tổng lập phương của các số lớn hơn 0.**

```
// Sokhong.cpp
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
{
    clrscr();
    n=16;
    cout<<"Ban nhap 16 phan tu cua mang ";
    for (j=1; j<=n; j++)
        {
            cout<<"a["<<j<<"]= ";
            cin>>a[j];
        }
```

```
        for (j=1; j<=n; j++)
            if (a[j]==0)
                m=m+1;
                cout<<"So so 0 co trong mang la: "<<m;
        getch();
        return 0;
}
```

**// Tongcacsoam.cpp**

```
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
{
        clrscr();
        n=16;
        cout<<"Ban nhap 16 phan tu cua mang ";
        for (j=1; j<=n; j++)
            {
                cout<<"a["<<j<<"]= ";
                cin>>a[j];
            }
        for (j=1; j<=n; j++)
            if (a[j]<0)
                m=m+a[j];
                cout<<"Tong cac so am co trong mang la:  " <<m;
        getch();
        return 0;
}
```

**// Tongcacsoduong.cpp**

```
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
{
```

```
    clrscr();
    n=16;
    cout<<"Ban nhap 16 phan tu cua mang ";
    for (j=1; j<=n; j++)
        {
          cout<<"a["<<j<<"]= ";
          cin>>a[j];
        }
    for (j=1; j<=n; j++)
        if (a[j]>0)
           m=m+a[j];
           cout<<"Tong cac so duong co trong mang la:  " <<m;
    getch();
    return 0;
}
```

**// Tongcacso.cpp**

```
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
{
    clrscr();
    n = 16;
    cout << " Ban nhap 16 phan tu cua mang ";
    for (j = 1;j<=n;j++)
        {
          cout<<"a["<<j<<"]= ";
          cin>>a[j];
        }
    for (j=1; j<=n; j++)
           m=m+a[j];
           cout<<"Tong cac so co trong mang la:  "<<m;
    getch();
    return 0;
}
```

```
// TongBinhPhuongcacsoam.cpp
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
{
      clrscr();
      n=16;
      cout<<"Ban nhap 16 phan tu cua mang ";
      for (j=1; j<=n; j++)
          {
             cout<<"a["<<j<<"]= ";
             cin>>a[j];
          }
      for (j=1; j<=n; j++)
          if (a[j]<0)
             m=m+a[j]*a[j];
             cout<<"Tong binh phuong cac so am co trong mang la:"<<m;
      getch();
      return 0;
}
// Tongbinhphuongcacsoduong.cpp
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
{
      clrscr();
      n=16;
      cout<<"Ban nhap 16 phan tu cua mang ";
      for (j=1; j<=n; j++)
          {
             cout<<"a["<<j<<"]= ";
             cin>>a[j];
```

```
            }
        for (j=1; j<=n; j++)
            if (a[j]>0)
              m=m+a[j]*a[j];
              cout<<"Tong binh phuong cac so duong co trong mang la: "<<m;
        getch();
        return 0;
}
```

**// Tonglapphuongcacsoam.cpp**

```
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
{
      clrscr();
      n=16;
      cout<<"Ban nhap 16 phan tu cua mang ";
      for (j=1; j<=n; j++)
          {
            cout << "a["<<j<<"] = ";
            cin >> a[j];
          }
      for (j=1; j<=n; j++)
          if (a[j]<0)
            m = m+a[j]*a[j]*a[j];
            cout << " Tong lap phuong cac so am co trong mang la: "<<m;
      getch();
      return 0;
}
```

**// Tonglapphuongcacsoduong.cpp**

```
#include <iostream.h>
#include <conio.h>
int j,m, n;
int a[17];
main()
```

```
{
    clrscr();
    n=16;
    cout<<"Ban nhap 16 phan tu cua mang ";
    for (j=1; j<=n; j++)
        {
            cout<<"a["<<j<<"]= ";
            cin>>a[j];
        }
    for (j=1; j<=n; j++)
        if (a[j]>0)
            m=m+a[j]*a[j]*a[j];
            cout<<"Tong lap phuong cac so duong co trong mang la:"<<m;
    getch();
    return 0;
}
```

## Xếp dãy tăng dần, giảm dần

⇨ Nhập vào mảng cố định một chiều 10 phần tử, chúng là các số nguyên, sau khi nhập xong, người sử dụng có thể thấy kết quả các phần tử được sắp xếp theo thứ tự tăng dần. Tương tự như vậy đối với chương trình dãy giảm, các phần tử sẽ sắp xếp giảm dần. Có vài cách sắp xếp, ở đây chúng tôi giới thiệu phương pháp Bubble Sort.

**// Daytang.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,x;
int a[11];
main()
{
    clrscr();
    cout<<"Ban nhap 10 phan tu cua mang ";
    for (j=1; j<=10; j++)
        {
            cout<<"a["<<j<<"]= ";
            cin>>a[j];
```

```
        }
    for (i=2; i<=10; i++)
        for (j=10; j>=i; j--)
          if (a[j-1]>a[j])
              {
                x=a[j-1] ;
                a[j-1]=a[j];
                a[j]=x;
              }
    cout<<"Day tang la ";
    for (i=1; i<=10; i++)
        cout<<a[i]<<"   ";
        getch();
        return 0;
}
```

**// Daygiam.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,x;
int a[11];
main()
{
    clrscr();
    cout << " Ban nhap 10 phan tu cua mang ";
    for (j=1; j<=10; j++)
        {
          cout << "a["<<j<<"] = ";
          cin>>a[j];
        }
    for (i=2; i<=10; i++)
        for (j=10; j>=i; j--)
          if (a[j-1]<a[j])
              {
                x=a[j-1] ;
                a[j-1]=a[j];
                a[j]=x;
```

```
                }
        cout<<"Day giam la ";
        for (i=1; i<=10; i++)
            cout<<a[i]<<"   ";
            getch();
            return 0;
}
```

⊙ Chạy thử chương trình Daytang và daygiam nhập 10 phần tử bất kỳ như sau: 4 2 5 8 34 56 78 56 58 89

Bạn sẽ thấy chương trình sắp xếp theo dãy tăng và giảm như sau

2 4 5 8 34 56 56 58 78 89

89 78 58 56 56 34 8 5 4 2

# Nhập một số phần tử bất kỳ xếp thành dãy tăng dần, giảm dần

**// Daytang.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,x,n;
int a[1000];
main()
{
    clrscr();
    cout<<"Ban nhap so phan tu cua mang n=  ";
    cin>>n;
    for (j=1; j<=n; j++)
        {
          cout<<"a[" <<j<<"]= ";
          cin>>a[j];
        }
    for (i=2; i<=n; i++)
        for (j=n; j>=i; j--)
          if (a[j-1]>a[j])
            {
                x=a[j-1];
```

```
                a[j-1]=a[j];
                a[j]=x;
            }
    cout<<"Day tang la  ";
    for (i=1; i<=n; i++)
        cout<<a[i]<<"   ";
        getch();
        return 0;
}
```

**// Daygiam.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,x,n;
int a[1000];
main()
{
    clrscr();
    cout << " Ban nhap so phan tu cua mang n =  ";
    cin >> n;
    for (j=1; j<=n; j++)
        {
            cout << "a[" <<j<<"] = ";
            cin >> a[j];
        }
    for (i=2; i<=n; i++)
        for (j=n; j>=i; j--)
            if (a[j-1]<a[j])
                {
                    x=a[j-1];
                    a[j-1]=a[j];
                    a[j]=x;
                }
    cout<<"Day giam la  ";
    for (i=1; i<=n; i++)
        cout<<a[i]<<"   ";
        getch();
```

```
        return 0;
}
```

⇨ Chương trình này cũng giống như trên, chúng tôi chỉ thay đổi miền của mảng có giá trị lớn hơn. Mảng nhập vào với số phần tử bất kỳ n <=1000. Chương trình này giải quyết vấn đề dùng mảng động mà chúng tôi đã đề cập ở những bài trước.

◎ Chạy thử chương trình Daytang, nhập số phần tử của mảng là 12 và giá trị của các phần tử được nhập theo thứ tự như sau:

1 2 3 56 45 2 3 4 1 5 6 7

Bạn sẽ có kết quả sắp xếp như sau:

Dãy tăng là:

1 1 2 2 3 3 4 5 6 7 45 56

◎ Chạy thử chương trình dãy giảm, nhập vào số phần tử của mảng là 15, giá trị của các phần tử theo thứ tự như sau:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Bạn sẽ có kết quả như sau

Dãy giảm là:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

# ✍ Nhập một số phần tử bất kỳ xếp thành dãy tăng dần, giảm dần

```
// Daytang.cpp
#include <iostream.h>
#include <conio.h>
int i,j,x,n;
int a[1000];
main()
{
    clrscr();
    cout<<"Ban nhap so phan tu cua mang n= ";
    cin>>n;
    for (j=1; j<=n; j++)
        {
            cout<<"a[" <<j<<"]= ";
            cin>>a[j];
```

```
        }
    for (i=2; i<=n; i++)
        for (j=n; j>=i; j--)
          if (a[j-1]>a[j])
            {
                x=a[j-1];
                a[j-1]=a[j];
                a[j]=x;
            }
    cout<<"Day tang la  ";
    for (i=1; i<=n; i++)
        cout << a[i] << "   ";
        getch();
        return 0;
}
```

**// Daygiam.cpp**

```
#include <iostream.h>
#include <conio.h>
int i,j,x,n;
int a[1000];
main()
{
    clrscr();
    cout << " Ban nhap so phan tu cua mang n =  ";
    cin >> n;
    for (j=1; j<=n; j++)
        {
          cout<<"a["<<j<<"]= ";
          cin>>a[j];
        }
    for (i=2; i<=n; i++)
        for (j=n; j>=i; j--)
          if (a[j-1]<a[j])
            {
                x=a[j-1];
                a[j-1]=a[j];
```

```
                a[j]= x;
            }
    cout<<"Day giam la  ";
    for (i=1; i<=n; i++)
        cout<<a[i]<<"   ";
        getch();
        return 0;
}
```

⇨ Chương trình này cũng giống như trên, chúng tôi chỉ thay đổi miền của mảng có giá trị lớn hơn. Mảng nhập vào với số phần tử bất kỳ n <=1000. Chương trình này giải quyết vấn đề dùng mảng động mà chúng tôi đã đề cập ở những bài trước.

◎ Chạy thử chương trình Daytang, nhập số phần tử của mảng là 12 và giá trị của các phần tử được nhập theo thứ tự như sau:

1 2 3 56 45 2 3 4 1 5 6 7

Bạn sẽ có kết quả sắp xếp như sau:

Dãy tăng là:

1 1 2 2 3 3 4 5 6 7 45 56

◎ Chạy thử chương trình dãy giảm, nhập vào số phần tử của mảng là 15, giá trị của các phần tử theo thứ tự như sau:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Bạn sẽ có kết quả như sau

Dãy giảm là:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1.

# ✍ Tổng và tích của hai ma trận 3x3

*Ví dụ 1:*

| Ma trận A | | | Ma trận B | | |
|---|---|---|---|---|---|
| 3 | 5 | 7 | 1 | 2 | 4 |
| 2 | -1 | 0 | 2 | 3 | -2 |
| 4 | 3 | 2 | -1 | 0 | 1 |

$$A + B = \begin{matrix} 3+1 & 5+2 & 7+4 \\ 2+2 & -1+3 & 0-2 \\ 4-1 & 3+0 & 2+1 \end{matrix} = \begin{matrix} 4 & 7 & 11 \\ 4 & 2 & -2 \\ 3 & 3 & 3 \end{matrix}$$

Như vậy muốn tính tổng của hai ma trận A và B ta lấy từng phần tử tương ứng của hai ma trận và cộng chúng lại với nhau.

*Ví dụ 2:*

Cho hai ma trận A và ma trận B như sau:

| Ma trận A | | | Ma trận B | | |
|---|---|---|---|---|---|
| 1 | 3 | 1 | 2 | 1 | 0 |
| 2 | 0 | 4 | 1 | -1 | 2 |
| 1 | 2 | 3 | 3 | 2 | 1 |

Tính tích của hai ma trận AxB và BxA

| | | | |
|---|---|---|---|
| | 1.2+3.1+1.3 | 1.1+3(-1)+1.2 | 1.0+3.2+1.1 |
| A x B = | 2.2+0.1+4.3 | 2.1+0(-1)+4.2 | 2.0+0.2+4.1 |
| | 1.2+2.1+3.3 | 1.1+2(-1)+3.2 | 1.0+2.2+3.1 |

| | | | |
|---|---|---|---|
| | 8 | 0 | 7 |
| A x B = | 16 | 10 | 4 |
| | 13 | 5 | 7 |

Nếu bạn tính tích BxA, bạn xem AxB có bằng BxA hay không?

| | | | |
|---|---|---|---|
| | 2.1+1.2+0.1 | 2.3+1.0 +0.2 | 2.1+1.4 +0.3 |
| B x A = | 1.1-1.2+2.1 | 1.3 -1.0 +2.2 | 1.1- 1.4 +2.3 |
| | 3.1+2.2+1.1 | 3.3+2.0 +1.2 | 3.1+2.4 +1.3 |

| | | | |
|---|---|---|---|
| | 4 | 6 | 6 |
| B x A = | 1 | 7 | 3 |
| | 8 | 11 | 14 |

Vậy luật giao hoán không thoả với tích của hai ma trận:

AxB<>BxA

```
// TongTichhaimatran.cpp
#include <iostream.h>
#include <conio.h>
int Mang[10][10];
int a[10][10];
int b[10][10];
int Tong[10][10];
int Tich[10][10];
int n;
int i,j,k;
main()
{
    clrscr();
    cout<<"Ban nhap so dong va so cot cua ma tran vuong n<10 "<<"\n";
```

```
    cin>>n;
    cout<<"Nhap cac phan tu cho ma tran a ";
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
              cout<<"Ban nhap a["<<i<<"]["<<j<<"]= ";
              cin>>a[i][j];
          }
    cout<<"Nhap cac phan tu cho ma tran b "<<"\n";
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
              cout<<"Ban nhap b["<<i<<"][" << j << "]= ";
              cin>>b[i][j];
          }
// Tinh Tong hai ma tran
  for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          Tong[i][j]=a[i][j]+b[i][j];
// Tinh Tich hai ma tran
  for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
            Tich[i][j]=0;
            for (k=1; k<=n; k++)
                Tich[i][j]=Tich[i][j]+a[i][k]*b[k][j];
          }
        cout<<"Ma tran a la: "<<"\n";
        for (i=1; i<=n; i++)
          for (j=1; j<=n; j++)
            {
              cout<<a[i][j]<<"     ";
              if (j==n)
                  cout<<"\n";
            }
        cout<<"Ma tran b la "<<"\n";
```

```
    for (i=1; i<=n; i++)
      for (j=1; j<=n; j++)
        {
            cout<<b[i][j]<<"     ";
            if (j==n)
              cout<<"\n";
        }
    cout << " Ma tran Tong la " << "\n";
    for (i=1; i<=n; i++)
      for (j=1; j<=n; j++)
       {
        cout << Tong[i][j] << "     ";
        if (j==n)
            cout << "\n";
       }
    cout << " Ma tran Tich la " << "\n";
    for (i=1; i<=n; i++)
      for (j=1; j<=n; j++)
        {
            cout << Tich[i][j] << "     ";
            if (j==n)
              cout<<"\n";
        }
    getch();
    return 0;
}
```

⇨ Trong bài trên, chúng tôi khai báo bốn mảng hai chiều, mảng a và b dùng để chứa hai mảng nhập vào. Mảng Tong và Tich dùng để chứa kết quả của Tổng và Tích của hai ma trận.

⇨ Đầu tiên bạn khai báo độ lớn của mảng.

⇨ Nhập các phần tử của ma trận a.

⇨ Kế đến là nhập các phần tử của ma trận b.

⇨ Tính tổng của hai ma trận vừa nhập và cho vào mảng có tên là Tong.

⇨ Tính tích của hai ma trận và cho vào mảng có tên là Tich.

⇨ In ra màn hình ma trận a, ma trận b, ma trận tổng và ma trận tích của hai ma trận nhập vào. Bạn cần có những khoảng trắng để phân cách các cột, điều kiện j = n sẽ xuống dòng cho đúng.

◎ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

◎ Giả sử bạn nhập vào mảng có độ lớn là 5. Chương trình sẽ cho bạn nhập hai ma trận a và b, sau khi nhập xong, chương trình sẽ in ra hai ma trận Tổng và Tích. Bạn có thể tham khảo khi chạy chương trình.

# In ma trận tam giác trên và ma trận tam giác dưới dạng 5x5

Bạn dùng ma trận hai chiều cố định gồm có 5 hàng 5 cột, để chứa ma trận vuông 5x5

*Ví dụ:* Nếu chúng ta có ma trận vuông 5x5 như sau:

```
1 2 3 4 5
3 5 6 7 8
3 4 9 6 3
5 3 5 7 8
8 4 2 6 9
```

Thì ma trận dưới sẽ là:

```
1
3 5
3 4 9
5 3 5 7
8 4 2 6 9
```

Ma trận trên sẽ là:

```
1 2 3 4 5
  5 6 7 8
    9 6 3
      7 8
        9
```

**// TamgiacTrenduoi.cpp**

```
#include <iostream.h>
#include <conio.h>
int a[10][10];
int n;
int i,j;
main()
{
    clrscr();
    cout<<"Ban nhap so dong va so cot cua ma tran vuong n <10 "<< "\n";
    cin >> n;
    cout << " Nhap cac phan tu cho ma tran a " << "\n";
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
              cout << " Ban nhap a[" << i << "][" << j << "] = ";
              cin >> a[i][j];
          }
  // Xac dinh ma tran tam giac duoi
   cout << " Ma tran tam giac duoi " << "\n";
   for (i=1; i<=n; i++)
         for (j=1; j<=i; j++)
          {
           cout<<a[i][j]<<"    ";
           if (j==i)
              cout<<"\n";
          }
   cout<<"Ma tran tam giac Tren "<<"\n";
   for (i=1; i<=n; i++)
    {
     for (j=1; j<=i-1; j++)
         cout<<"    ";
     for (j=i; j<=n; j++)
          cout<<a[i][j]<<"    ";
          cout<<"\n";
    }
  getch();
```

```
    return 0;

}
```

⇨ Bài này tương đối đơn giản, ta chỉ dùng các thao tác duyệt mảng hai chiều.

⇨ Khi in ma trận tam giác dưới, chúng tôi dùng hai vòng for, lưu ý rằng vòng for bên trong chỉ thực hiện từ 1 tới i, và vì vậy một nửa ma trận dưới được in ra.

⇨ Việc thực hiện ma trận tam giác trên, vòng for ở giữa thực hiện in ra khoảng trắng để các phần tử của nửa trên được in ra đúng vị trí.

◎ Chạy thử và nhập vào một ma trận vuông bất kỳ, bạn sẽ thấy ma trận trên và ma trận dưới được in ra.

# ☎ Phần 6

# Bài tập sử dụng THỦ TỤC và HÀM

## ✍ Dùng thủ tục và hàm để tính n!

Bây giờ các bạn viết chương trình nhập vào một số nguyên bất kỳ để tính giai thừa của nó. Trong chương trình này, các bạn phải dùng hàm hoặc thủ tục để tính toán.

Nếu người sử dụng nhập vào 0 thì chương trình thông báo 0!=1

Nếu nhập vào một số âm thì thoát khỏi chương trình.

Nếu nhập số lớn hơn 0 thì sẽ tính giai thừa đúng theo định nghĩa.

Bài này chúng tôi sẽ giới thiệu cho các bạn dùng chương trình con. Chương trình con là một đoạn chương trình có tên, chúng sẽ được gọi để thực hiện ở nhiều nơi trong chương trình. Nơi gọi có thể cung cấp các thông số (*Parameter*) cho chương trình con, và chương trình con có thể làm thay đổi giá trị của một vài trong số các thông số đó.

Các bạn sẽ dùng chương trình con khi:

- Có một công việc được thực hiện nhiều nơi (*ở mỗi nơi có thông số cung cấp và kết quả riêng, nhưng bản chất công việc không thay đổi*).
- Khi các bạn cần phân đoạn chương trình để chương trình trong sáng và dễ hiểu.

Có hai loại chương trình con, đó là thủ tục (*Procedure*) và hàm (*Function*)

Chúng khác nhau: sự gọi thủ tục là một phát biểu, còn sự gọi hàm là một phép toán, là một thành phần của biểu thức, có sự trả về một kết quả, thủ tục thì không.

### ☞ Viết bằng thủ tục

```
// GiaithuaThutuc.cpp
#include <iostream.h>
#include <conio.h>
void giaithua(int,float&);
```

```
main()
{
  int i;
  int n;
  float x;
  cout<<"Ban nhap mot so de tinh giai thua:  ";
  cin>>n;
    if (n>=0)
          {
             giaithua(n, x);
             cout<<n<<"!= "<<x;
          }
    else
             cout<<"Ban khong duoc nhap so am. ";
  getch();
  return 0;
}
void giaithua(int m, float& gt)
  {
        int i;
        if (m==0)
          gt=1;
        else
        gt=1;
        for (i=1; i<=m; i++)
           gt=gt*i;
  }
```

⇨ Chương trình trên sẽ cho bạn nhập vào một số để tính giai thừa, dùng thủ tục, tức là phải dùng hàm void. Chúng ta phải khai báo như sau:

```
void Ten_ham (...)
```

⇨ Hàm này không trả về một trị có kiểu đã biết như char, int, float... nào cả.

⇨ Khi bạn nhập vào một số nguyên dương hoặc bằng 0, bạn sẽ có kết quả giai thừa của số đó, nếu bạn nhập số âm, sẽ thông báo lỗi sai.

☞ **Viết bằng hàm**

```
// GiaithuaHam.cpp
#include <iostream.h>
#include <conio.h>
long int giaithua(int,long int);
main()
{
   int i;
   int n;
   float x;
   cout<<"Ban nhap mot so de tinh giai thua: ";
   cin>>n;
   if (n<0)
         cout<<"Ban khong duoc nhap so am. ";
   else
         cout<<n<<"!= "<<giaithua(n,x);
  getch();
  return 0;
}
long int giaithua(int m, long int gt)
   {
         int i;
         if (m==0)
               gt = 1;
         else
              gt = 1;
            for (i=1;i<=m;i++)
              gt=gt*i;
            return(gt);
   }
```

⇨ Bạn nhìn hàm và thủ tục khác nhau, sự gọi hàm là một phép toán chứ không phải là một phát biểu.

⇨ Khi thân của hàm kết thúc thì ta dùng hàm return để trả về giá trị cho hàm là return(gt).

⇨ Dùng chương trình con rất là hay, giúp cho chương trình sáng sủa hơn, bạn cần nghiên cứu để phân biệt giữa thủ tục và hàm.

⊙ Nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Bạn thấy kết quả thông báo rất chính xác. Bạn thử chạy và cho một số trường hợp qua các giá trị đặc biệt của n.

# ✍ Dùng thủ tục và hàm để tính bốn phép toán cơ bản

Để luyện tập thêm về cách dùng thủ tục và hàm, chúng tôi cho các bạn làm quen bằng một bài tập thật đơn giản, đó là làm bốn phép toán cơ bản.

Yêu cầu của chương trình là thiết kế sao cho người sử dụng nhập vào 2 số nguyên a và b. Bạn có thể xem kết quả của:

1- a+b
2- a–b
3- axb
4- a/b
5- a mũ b

☞ **Viết bằng thủ tục**

```
// PheptoandungThutuc.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
float a,b;
float c;
void Cong(float x, float y, float& z)
{
    z=x+y;
}
void Tru(float x, float y, float& z)
{
    z=x-y;
}
```

```
void Nhan(float x, float y, float& z)
{
    z=x*y;
}
void Chia(float x, float y, float& z)
{
    z=x/y;
}
void Luythua(float x, float y, float& z)
{
    z=pow(x, y);
}
main()
{
    clrscr();
    cout<<"Ban nhap vao so a =  ";
    cin>>a;
    cout<<"\n Ban nhap vao so b=  ";
    cin>>b;
    Cong(a,b,c);
    cout<<a<<"+"<<b<<"="<<c<<"\n";
    Tru(a,b,c);
    cout<<a<<"-"<<b<<"="<<c<<"\n";
    Nhan(a,b,c);
    cout << a << " x "<< b << " =  "<< c << "\n";
    Chia(a,b,c);
    cout << a << " / "<< b << " =  "<< c << "\n";
    Luythua(a,b,c);
    cout << a << " luy thua "<< b << " =  "<< c << "\n";
    getch();
    return 0;
}
```

☞ **Viết bằng Hàm**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
```

```
float a,b;
float c;

float Cong(float x, float y, float z)
{
    z = x+y;
    return(z);
}
float Tru(float x, float y, float z)
{
    z = x-y;
    return(z);
}
float Nhan(float x, float y, float z)
{
    z = x*y;
    return(z);
}
float Chia(float x, float y, float z)
{
    z=x/y;
    return(z);
}
float Luythua(float x, float y, float z)
{
    z=pow(x, y);
    return(z);
}
main()
{
    clrscr();
    cout<<"Ban nhap vao so a= ";
    cin>>a;
    cout<<"\n Ban nhap vao so b= ";
    cin>>b;
    cout<<a<<"+"<<b<<"="<<Cong(a,b,c)<<"\n";
```

```
    cout<<a<<"-"<<b<<"="<<Tru(a,b,c)<<"\n";
    cout<<a<<"x"<<b<<"="<<Nhan(a,b,c)<<"\n";
    cout<<a<<"/"<<b<<"="<<Chia(a,b,c)<<"\n";
    cout<<a<<"luy thua"<<b<<"="<<Luythua(a,b,c)<<"\n";
    getch();
    return 0;
}
```

⇨ Trong cách tính bốn phép toán đơn giản, chúng tôi dùng thủ tục và dùng hàm. Bạn xem lại để nắm vững cách viết một chương trình khi dùng thủ tục, cách viết một chương trình khi dùng hàm.

*Lưu ý:*

Dùng hàm thì luôn luôn có giá trị trả về.

⊙ Chạy thử chương trình viết bằng thủ tục hoặc bằng hàm ở trên, nhập giá trị a=2 và b=3, bạn sẽ thấy kết quả như sau:

```
2+3=5
2-3=-1
2x3=6
2/3=0.666667
2 luy thua 3=8
```

# ✍ In ra ước số chung của a và b

☞ **Viết bằng cách dùng Hàm**

```
// USCLN_ham.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int uscln(int x, int y)
{
    int t;
    x = abs(x);
    y = abs(y);
    do
      {
        t = y;
        y = (x%y);
```

```
            x=t;
        }
    while (y!=0);
    return (x);
}
main()
{
    int n,m;
    clrscr();
    cout<<"Ban nhap vao so thu nhat n = ";
    cin>>n;
    cout<<"Ban nhap vao so thu hai m = ";
    cin>>m;
    cout<<"Uoc so chung lon nhat cua "<<n<<"va "<<m<<"la: "<<uscln(n,m);
    getch();
    return 0;
}
```

Bạn cũng có thể viết như sau:

**// USCLN_ham.cpp**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
int uscln(int x, int y)
{
    int t;
    x=abs(x);
    y=abs(y);
    do
      {
        if (x>y)
          x=x-y;
        if (y>x)
          y=y-x;
      }
    while (x!=y);
    return(x);
```

```
}
main()
{
    int n,m;
    clrscr();
    cout<<"Ban nhap vao so thu nhat n =  ";
    cin>>n;
    cout<<"Ban nhap vao so thu hai m=   ";
    cin>>m;
    cout<<"Uoc so chung lon nhat cua "<<n<<"va"<<m<<"la:"<<uscln(n,m);
    getch();
    return 0;
}
```

Trong chương trình trên, bạn dùng chương trình con là hàm để tính USCLN của hai số.

Dùng vòng lặp mới DO ... WHILE <ĐK>. Vòng lặp này sẽ kết thúc khi biểu thức điều kiện sai.

Chúng tôi viết theo hai dạng, bạn có thể tham khảo để biết thêm.

Bây giờ chúng tôi viết bằng cách dùng thủ tục. Bạn thấy chương trình sẽ như sau:

**// USCLN_Thutuc.cpp**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
void uscln(int x, int y, int& t)
{
    x = abs(x);
    y = abs(y);
    do
      {
        t = y;
        y = (x%y);
        x = t;
      }
    while(y!=0);
}
```

```
main()
{
    int n,m,a;
    clrscr();
    cout<<"Ban nhap vao so thu nhat n=  ";
    cin>>n;
    cout<<"Ban nhap vao so thu hai m=  ";
    cin>>m;
    uscln(n,m,a);
    cout<<"Uoc so chung lon nhat cua "<<n<<" va "<<m<<" la: "<<a;
    getch();
    return 0;
}
```

Bạn có thể viết lại như sau:

```
// USCLN_Thutuc.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
void uscln(int x, int y, int&t)
{
    x=abs(x);
    y=abs(y);
    do
      {
        if (x>y)
          x=x-y;
        if (y>x)
          y=y-x;
      }
    while (x!=y);
}
main()
{
   int n,m,a;
   clrscr();
   cout<<"Ban nhap vao so thu nhat n=  ";
```

```
    cin>>n;
    cout<<"Ban nhap vao so thu hai m=  ";
    cin>>m;
    uscln(n,m,a);
    cout<<"Uoc so chung lon nhat cua "<<n<<" va "<<m<<"la: "<<a;
    getch();
    return 0;
}
```

# Cho biết một phân số có tối giản hay không

```
// Phan_so_toi_gian.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int Tuso, Mauso;
int USCLN(int x, int y)
{
      int t;
      x=abs(x);
      y=abs(y);
      do
          {
            t= y;
            y= x%y;
            x= t;
          }
      while (y!=0);
      return (x);
}
main()
{
    cout << "Ban nhap vao tu so :  ";
    cin >>Tuso;
```

```
    cout<<"Ban nhap vao mau so:  ";
    cin>>Mauso;
    if (USCLN(Tuso,Mauso)==1)
      cout<<"Phan so ban vua nhap la phan so toi gian ";
    else
      cout<<"Phan so ban vua nhap khong phai la phan so toi gian ";
    getch();
    return 0;
}
```

⇨ Chương trình này cũng khá giống chương trình trên vì phải tìm ước số chung lớn nhất của hai số (tử số và mẫu số).

⇨ Bạn chỉ cần thêm một phát biểu If để kiểm tra điều kiện của USCLN đó, nếu USCLN(Tuso,Mauso)=1 thì đó là phân số tối giản.

⇨ Các bạn có thể viết chương trình bằng cách dùng thủ tục thay cho hàm.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Bạn nhập tử số=1 và mẫu số=4, bạn sẽ thấy chương trình báo đây là phân số tối giản.

⊙ Nếu bạn nhập tử số=2 và mẫu số=8, bạn sẽ thấy chương trình báo đây không phải là phân số tối giản.

# Báo ra số lớn nhất trong hai số

**Dùng hàm**

```
// Max_cuahaiso_DungHam.cpp
#include <iostream.h>
#include <conio.h>
float a,b;
float Max(float m, float n)
{
  if (m>n)
    return(m);
  else
    return(n);
}
```

```
main()
{
    cout<<"Ban nhap vao so a:  ";
    cin>>a;
    cout<<"Ban nhap vao so b:  ";
    cin>>b;
    cout<<"So lon nhat cua hai so tren la: "<<Max(a,b);
    getch();
    return 0;
}
```

☞ **Dùng Thủ tục**

```
// Max_cuahaiso_DungThuTuc.cpp
#include <iostream.h>
#include <conio.h>
float a,b;
float x;
void Max(float m, float n, float& z)
{
  if (m>n)
      z=m;
  else
      z = n;
}
main()
{
    cout << " Ban nhap vao so a :  ";
    cin >> a;
    cout << " Ban nhap vao so b :  ";
    cin >> b;
    Max(a,b,x);
    cout << " So lon nhat cua hai so tren la : " << x;
    getch();
    return 0;
}
```

⊙ Nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Bạn nhập vào hai số 45 và 76, bạn sẽ được kết quả như sau:
Số lớn nhất của hai số trên là: 76

## Báo ra số lớn nhất trong ba số

```
// Max_cuabaso_DungHam.cpp
#include <iostream.h>
#include <conio.h>
float a,b,c;
float Max(float m, float n)
{
  if (m>n)
    return(m);
  else
    return(n);
}
main()
{
    clrscr();
    cout<<"Ban nhap vao so a:  ";
    cin>>a;
    cout<<"Ban nhap vao so b:  ";
    cin>>b;
    cout<<"Ban nhap vao so c:  ";
    cin>>c;
    cout<<"So lon nhat cua ba so tren la: "<<Max(a, Max(b,c));
    getch();
    return 0;
}
```

Chương trình này, chúng tôi dùng hàm tính Max của hai số sau đó áp dụng nó để tính được Max của ba số theo biểu thức *Max(a, Max(b,c))*.

Bạn có thể áp dụng để tính được Max của nhiều số, dưới đây chúng tôi viết thêm Max của bốn số, năm số và mười số. Các bạn có thể viết thêm cách tính Max của nhiều số một cách dễ dàng, chỉ cần thêm một số biến trung gian để chứa kết quả của các phép so sánh một số lượng số ít hơn.

```
// Max_cuabonso_DungHam.cpp
#include <iostream.h>
```

```
#include <conio.h>
float a,b,c,d;
float Max(float m, float n)
{
   if (m>n)
     return(m);
   else
     return(n);
}
main()
{
     clrscr();
     cout<<"Ban nhap vao so a:  ";
     cin>>a;
     cout<<"Ban nhap vao so b:  ";
     cin>>b;
     cout<<"Ban nhap vao so c:  ";
     cin>>c;
     cout<<"Ban nhap vao so d:  ";
     cin >> d;
     cout <<"So lon nhat cua bon so tren la:"<< Max(Max(a,b), Max(c,d) ) ;
     getch();
     return 0;
}
// Max_cuanamso_DungHam;
#include <iostream.h>
#include <conio.h>
float a,b,c,d,e;
float Ketqua;
float  Max(float m, float n)
{

  if (m>n)
    return(m);
  else
    return(n);
```

```
}
main()
{
    clrscr();
    cout<<"Ban nhap vao so a:  ";
    cin>>a;
    cout<<"Ban nhap vao so b:  ";
    cin>>b;
    cout<<"Ban nhap vao so c:  ";
    cin>>c;
    cout<<"Ban nhap vao so d:  ";
    cin>>d;
    cout<<"Ban nhap vao so e:  ";
    cin>>e;
    Ketqua=Max(Max(a,b),Max(c,d));
    cout<<"So lon nhat cua nam so tren la: "<<Max(Ketqua, e);
    getch();
    return 0;
}
```

**// Max_cuamuoiso_DungHam.cpp**

```
#include <iostream.h>
#include <conio.h>
float a,b,c,d,e,f,g,h,i,j;
float Ketqua1;
float Ketqua2;
float Max(float m, float n)
{
  if (m>n)
    return(m);
  else
    return(n);
}
main()
{
    clrscr();
    cout<<"Ban nhap vao so a:  ";
```

```
    cin>>a;
    cout<<"Ban nhap vao so b:  ";
    cin>>b;
    cout<<"Ban nhap vao so c:  ";
    cin>>c;
    cout<<"Ban nhap vao so d:  ";
    cin>>d;
    cout<<"Ban nhap vao so e:  ";
    cin>>e;
    cout<<"Ban nhap vao so f:  ";
    cin>>f;
    cout<<"Ban nhap vao so g:  ";
    cin>>g;
    cout<<"Ban nhap vao so h:  ";
    cin>>h;
    cout<<"Ban nhap vao so i:  ";
    cin>>i;
    cout<<"Ban nhap vao so j:  ";
    cin>>j;
    Ketqua1=Max(Max(a,b),Max(c,d));
    Ketqua2= Max(Max(e,f),Max(g,h));
    cout << " So lon nhat cua muoi so tren la : "<<
            Max(Max(Ketqua1,Ketqua2),Max(i,j));
    getch();
    return 0;
}
```

## Báo ra phân số tối giản của một phân số được nhập vào

```
// Uocluong_phanso.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int n;
```

```
int Tuso,Mauso;
int USCLN(int x, int y)
{
      int t;
      x=abs(x);
      y=abs(y);
      do
          {
            t=y;
            y=x%y;
            x=t;
          }
      while (y!=0);
      return(x);
}
main()
{
   clrscr();
   cout<<"Ban nhap vao tu so= ";
   cin>>Tuso;
   cout<<"Ban nhap vao mau so= ";
   cin>>Mauso;
   if (Mauso!=0)
      {
          if (Tuso!=0)
            {
              {
                    n=USCLN(Tuso,Mauso);
                    Tuso=Tuso/n;
                    Mauso=Mauso/n;
              }
            if (Mauso<0)
              {
                  Tuso=-Tuso;
                  Mauso=-Mauso;
              }
```

```
        }
      else
        Mauso=1;
        cout<<"Phan so toi gian la: "<<Tuso<<"/"<<Mauso;
    }
  else
    cout<<"Mau so khong duoc bang 0 ";
  getch();
  return 0;
}
```

⇨ Như bạn đã biết, *để có phân số tối giản, ta chia tử và mẫu của nó cho ước số chung lớn nhất của tử và mẫu.*

⇨ Với ví dụ trên, phân số p=2/8 có tử số là 2, mẫu số là 8, và ước số chung lớn nhất của 2 và 8 là 2.

⇨ Muốn có phân số tối giản của phân số trên, bạn chỉ cần chia tử và mẫu cho ước số chung lớn nhất của nó là 2, bạn sẽ được phân số tối giản là bằng 1/4.

⇨ Từ kết quả trên, bạn phải viết một hàm để tìm ước số chung lớn nhất của 2 số, hàm này các bạn đã viết rồi.

⇨ Các bạn cần phải chú ý và thực hiện đầy đủ các yêu cầu sau đây.

*Ví dụ:*

(-5)/4 và 5/(-4)
5/4 và (-5)/(-4)

⇨ Nếu một phân số nhỏ hơn 0, thì dấu sẽ nằm trên tử số

*Ví dụ:* 5/(-4) thì phải báo ra là –5/4
(-5)/(-4)=5/4

⇨ Nếu tử số khác không thì tiến hành tìm phân số tối giản như đã hướng dẫn ở trên, đồng thời chuyển dấu phân số cho chính xác. Ngược lại (tử số=0) thì mẫu số lúc này phải bằng bao nhiêu (bạn chỉ cần tìm USCLN của 0 và giá trị của mẫu số), sau đó thực hiện phép chia như trên để biết giá trị của mẫu số lúc này.

*Ghi chú:* Số 0 chia hết cho mọi số (khác 0) đều bằng 0.

⊙ Bạn nhấn Ctrl+F9 để chạy thử chương trình.

⊙ Nhập tử số=5 , mẫu số=-15. Bạn sẽ có kết quả như sau

Phân số tối giản là –1/3

⦿ Bạn thấy đã chuyển dấu lên tử số.

⦿ Nếu tử số=8 và mẫu số=16 thì bạn sẽ có thông báo

Phân số tối giản là 1/2

⦿ Bạn thử thêm một số ví dụ nữa xem sao.

# ✍ Nhập vào hai phân số, tính

## ❑ Tổng của hai phân số

## ❑ Hiệu của hai phân số

## ❑ Tích của hai phân số

## ❑ Thương của hai phân số

⇨ Trước khi tính tổng, hiệu, tích, thương của hai phân số, phân số đó phải là phân số tối giản *(nếu một phân số mà tử số bằng 0, thì mẫu số của nó sẽ bằng 1, đúng với phân số tối giản)* như chúng tôi đã giới thiệu ở bài trước.

⇨ Kết quả tính cuối cùng sẽ để ở dạng phân số tối giản, không tính ra kết quả là một số nguyên hay là một số thực.

⇨ Vậy trong thủ tục tính tổng hai phân số sẽ tiến hành theo các bước sau đây.

⇨ Ước lượng phân số thứ nhất, đồng thời đổi dấu cho đúng, xét trường hợp khi tử số bằng 0. Tương tự như vậy đối với phân số thứ hai.

⇨ Khi đã có hai phân số tối giản với dấu đã hợp lệ, chúng ta phải quy đồng mẫu số. Việc quy đồng mẫu số để tìm bội số chung nhỏ nhất, chúng tôi sẽ đề cập trong phần viết code.

⇨ Sau khi quy đồng mẫu số xong, tính tử số của phân số cần tính.

⇨ Khi đã có phân số là tổng của hai phân số trên, kết quả lúc này có thể phân số chưa là tối giản, chúng ta phải đưa qua bước tối giản. Bạn lưu ý trong Form khi sử dụng TextBox việc sử dụng biến cùng các phép toán và gọi hàm hơi phức tạp, bạn phải khôn khéo để giải quyết vấn đề để mang lại kết quả chính xác.

⇨ Tương tự như vậy đối với việc tính hiệu hai phân số, chỉ khác với thủ tục tính tổng một chút, đó là các bạn đổi giá trị của "tử số 2" thành "– tử số 2"

⇨ Đối với phép tính nhân hai phân số, thì không cần quy đồng mẫu số, chỉ cần thay đổi phần tử số và mẫu số ở phân số kết quả. Ta biết muốn nhân một phân số cho một phân số, ta nhân tử với tử, mẫu với mẫu.

⇨ Đối với phép chia một phân số cho một phân số, ta nhân phân số thứ nhất với phân số thứ hai đảo ngược.

```
// Tong.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int n,m,k;
int Tuso1,Tuso2,Mauso1,Mauso2;
int Tu,Mau;
int USCLN(int x, int y)
{
    int t;
    x=abs(x);
    y=abs(y);
    do
        {
          t= y;
          y= x%y;
          x= t;
        }
    while (y!=0);
    return (x);
}
main()
{
    clrscr();
    cout<<"Ban nhap tu so 1. Tuso1= ";
    cin>>Tuso1;
    cout<<"Ban nhap mau so 1. Mauso1= ";
    cin>>Mauso1;
    cout<<"Ban nhap tu so 2. Tuso2= ";
    cin>>Tuso2;
    cout<<"Ban nhap mau so 2. Mauso2= ";
```

```
cin>>Mauso2;
if (Tuso1!=0)
    {
        {
            n=USCLN(Tuso1,Mauso1);
            Tuso1=Tuso1/n;
            Mauso1=Mauso1/n;
        }
        if (Mauso1<0)
          {
              Tuso1=-Tuso1;
              Mauso1=-Mauso1;
          }
    }
else
      Mauso1=1;
if (Tuso2!=0)
    {
        {
            m=USCLN(Tuso2,Mauso2);
            Tuso2=Tuso2/m;
            Mauso2=Mauso2/m;
        }
        if (Mauso2<0)
          {
              Tuso2=-Tuso2;
              Mauso2=-Mauso2;
          }
    }
else
      Mauso2=1;
Mau=( Mauso1/USCLN(Mauso1,Mauso2))*Mauso2;
Tu=Tuso1*(Mau/Mauso1)+Tuso2*(Mau/Mauso2);
if (Tu!=0)
    {
        {
```

```
                k=USCLN(Tu,Mau);
                Tu=Tu/k;
                Mau=Mau/k;
            }
            if (Mau<0)
                {
                  Tu=-Tu;
                  Mau=-Mau;
                }
        }
    else
        Mau=1;
        cout << "Tong hai phan so tren la : "<< Tu << " / "<< Mau ;
  getch();
}
```

**// Hieu.cpp**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
int n,m,k;
int Tuso1,Tuso2,Mauso1,Mauso2;
int Tu,Mau;
int USCLN(int x, int y)
{
    int t;
    x=abs(x);
    y=abs(y);
    do
        {
          t=y;
          y=x%y;
          x=t;
        }
    while (y!=0);
    return (x);
}
```

```
main()
{
    clrscr();
    cout<<"Ban nhap tu so 1. Tuso1= ";
    cin>>Tuso1;
    cout<<"Ban nhap mau so 1. Mauso1= ";
    cin>>Mauso1;
    cout<<"Ban nhap tu so 2. Tuso2= ";
    cin>>Tuso2;
    cout<<"Ban nhap mau so 2. Mauso2= ";
    cin>>Mauso2;
    if (Tuso1!=0)
        {
            {
                n=USCLN(Tuso1,Mauso1);
                Tuso1=Tuso1/n;
                Mauso1=Mauso1/n;
            }
            if (Mauso1<0)
              {
                  Tuso1=-Tuso1;
                  Mauso1=-Mauso1;
              }
        }
    else
         Mauso1=1;
    if (Tuso2!=0)
        {
          {
              m=USCLN(Tuso2,Mauso2);
              Tuso2=Tuso2/m;
              Mauso2=Mauso2/m;
          }
          if (Mauso2<0)
            {
                Tuso2=-Tuso2;
```

```
                    Mauso2=-Mauso2;
                }
            }
        else
            Mauso2=1;
        Mau=(Mauso1/USCLN(Mauso1,Mauso2))*Mauso2 ;
        Tu=Tuso1*(Mau/Mauso1)-Tuso2*(Mau/Mauso2);
        if (Tu!=0)
            {
              {
                    k = USCLN(Tu,Mau);
                    Tu = Tu/k;
                    Mau = Mau/k;
              }
              if (Mau<0)
                    {
                      Tu = -Tu;
                      Mau = -Mau;
                    }
            }
        else
            Mau = 1;
            cout<<"Hieu hai phan so tren la: "<<Tu<<"/"<<Mau;
    getch();
}
// Tich.cpp
#include <iostream.h>
#include <conio.h>
#include <math.h>
int n,m,k;
int Tuso1,Tuso2,Mauso1,Mauso2;
int Tu,Mau;
int USCLN(int x, int y)
{
    int t;
    x=abs(x);
```

```
        y=abs(y);
        do
            {
              t=y;
              y=x%y;
              x=t;
            }
        while (y!=0);
        return (x);
}
main()
{
    clrscr();
    cout<<"Ban nhap tu so 1. Tuso1= ";
    cin>>Tuso1;
    cout<<"Ban nhap mau so 1. Mauso1= ";
    cin>>Mauso1;
    cout<<"Ban nhap tu so 2. Tuso2= ";
    cin>>Tuso2;
    cout<<"Ban nhap mau so 2. Mauso2= ";
    cin>>Mauso2;
    if (Tuso1!=0)
        {
            {
                n=USCLN(Tuso1,Mauso1);
                Tuso1=Tuso1/n;
                Mauso1=Mauso1/n;
            }
            if (Mauso1<0)
              {
                Tuso1=-Tuso1;
                Mauso1=-Mauso1;
              }
        }
    else
        Mauso1=1;
```

```
    if (Tuso2!=0)
        {
          {
                m=USCLN(Tuso2,Mauso2);
                Tuso2=Tuso2/m;
                Mauso2=Mauso2/m;
          }
          if (Mauso2<0)
            {
                Tuso2 = -Tuso2;
                Mauso2 = -Mauso2;
            }
        }
    else
        Mauso2 = 1;
    Mau = Mauso1*Mauso2;
    Tu = Tuso1*Tuso2;
    if (Tu!=0)
        {
          {
                k = USCLN(Tu,Mau);
                Tu = Tu/k;
                Mau=Mau/k;
          }
          if (Mau<0)
                {
                  Tu=-Tu;
                  Mau=-Mau;
                }
        }
    else
        Mau=1;
        cout<<"Tich hai phan so tren la: "<<Tu<<"/"<<Mau;
  getch();
}
// Thuong.cpp
```

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
int n,m,k;
int Tuso1,Tuso2,Mauso1,Mauso2;
int Tu,Mau;
int USCLN(int x, int y)
{
      int t;
      x=abs(x);
      y=abs(y);
      do
          {
            t=y;
            y=x%y;
            x=t;
          }
      while (y!=0);
      return (x);
}
main()
{
     clrscr();
     cout<<"Ban nhap tu so 1. Tuso1=  ";
     cin>>Tuso1;
     cout<<"Ban nhap mau so 1. Mauso1=  ";
     cin>>Mauso1;
     cout<<"Ban nhap tu so 2. Tuso2=  ";
     cin>>Tuso2;
     cout<<"Ban nhap mau so 2. Mauso2=  ";
     cin>>Mauso2;
     if (Tuso1!=0)
         {
            {
                  n=USCLN(Tuso1,Mauso1);
                  Tuso1=Tuso1/n;
```

```
                Mauso1=Mauso1/n;
            }
            if (Mauso1<0)
              {
                  Tuso1=-Tuso1;
                  Mauso1=-Mauso1;
              }
        }
    else
        Mauso1=1;
    if (Tuso2!=0)
        {
          {
              m = USCLN(Tuso2,Mauso2);
              Tuso2 = Tuso2/m;
              Mauso2 = Mauso2/m;
          }
          if (Mauso2<0)
            {
                Tuso2 = -Tuso2;
                Mauso2 = -Mauso2;
            }
        }
    else
        Mauso2=1;
    Mau=Mauso1*Tuso2;
    Tu=Tuso1*Mauso2;
    if (Tu!=0)
        {
          {
              k=USCLN(Tu,Mau);
              Tu=Tu/k;
              Mau=Mau/k;
          }
          if (Mau<0)
              {
```

```
                Tu=-Tu;
                Mau=-Mau;
            }
        }
    else
        Mau=1;
        cout<<"Thuong hai phan so tren la: "<<Tu<<"/"<<Mau;
    getch();
}
```

Trong bài này các bạn biết thêm về bội số chung nhỏ nhất.

⊙ Nhấn Ctrl+F9 để chạy thử chương trình.

Bạn nhập Tử số 1=1
Mẫu số 1=3
Tử số 2=1
Mẫu số 2=2

⊙ Bạn sẽ có kết quả như sau:

Tổng của hai phân số trên là 5/6
Hiệu của hai phân số trên là –1/6
Tích của hai phân số trên là 1/6
Thương của hai phân số trên là 2/3

# ✍ Tích một số với ma trận Bình phương và lập phương của một ma trận

*Hướng dẫn*

Tích của một số m với ma trận A là ma trận xác định bởi đẳng thức.

$$m \times \begin{matrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{matrix} = \begin{matrix} ma11 & ma12 & ma13 \\ ma21 & ma22 & ma23 \\ ma31 & ma32 & ma33 \end{matrix}$$

+ bình phương A=AxA
+ Lập phương $A=A^2xA$

*Ghi chú:*

$A^2xA$ có kết quả khác $AxA^2$

**// Binhphuongmatran.cpp**

```
#include <iostream.h>
#include <conio.h>
int a[10][10];
int Binhphuong[10][10];
int n;
int i,j,k;
main()
{
    clrscr();
    cout<<"Ban nhap so dong va so cot cua ma tran vuong n <10 "<< "\n";
    cin >> n;
    cout << " Nhap cac phan tu cho ma tran a " << "\n";
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
              cout << " Ban nhap a[" << i << "][" << j << "] = ";
              cin >> a[i][j];
          }
   // Binh phuong cua ma tran a
     for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
             Binhphuong[i][j]=0;
             for (k=1; k<=n; k++)
                  Binhphuong[i][j]=Binhphuong[i][j]+a[i][k]*a[k][j];
          }
        cout<<"Ma tran a: "<<"\n";
        for (i=1; i<=n; i++)
          for (j=1; j<=n; j++)
            {
                cout<<a[i][j]<<"   ";
                if (j==n)
                   cout<<"\n";
            }
        cout<<"Binh phuong cua ma tran a "<<"\n";
        for (i=1; i<=n; i++)
```

```
            for (j =1; j<=n; j++)
              {
                cout<<Binhphuong[i][j]<<"   ";
                if (j==n)
                  cout<<"\n";
              }
        getch();
        return 0;
}
// Lapphuongmatran.cpp
#include <iostream.h>
#include <conio.h>
int a[10][10];
int Binhphuong[10][10];
int Lapphuong[10][10];
int n;
int i,j,k;
main()
{
    clrscr();
    cout<<"Ban nhap so dong va so cot cua ma tran vuong n <10 "<< "\n";
    cin>>n;
    cout<<"Nhap cac phan tu cho ma tran a "<<"\n";
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
            cout<<"Ban nhap a["<<i<<"]["<<j<<"]= ";
            cin>>a[i][j];
          }
   // Binh phuong cua ma tran a
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
            Binhphuong[i][j]=0;
            for (k=1; k<=n; k++)
              Binhphuong[i][j]=Binhphuong[i][j]+a[i][k]*a[k][j];
```

```
        }
// Lap phuong cua ma tran a
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
          {
            Lapphuong[i][j]=0;
            for (k=1; k<=n; k++)
              Lapphuong[i][j]=Lapphuong[i][j]+Binhphuong[i][k]*a[k][j];
          }
        cout << " Ma tran a : " << "\n";
        for (i=1; i<=n; i++)
          for (j=1; j<=n; j++)
            {
              cout << a[i][j] << "   ";
              if (j==n)
                cout << "\n";
            }
        cout << "Lap phuong cua ma tran a "<< "\n";
        for (i=1; i<=n; i++)
          for (j=1; j<=n; j++)
            {
              cout<<Lapphuong[i][j]<<"   ";
              if (j==n)
                cout<<"\n";
            }
        getch();
        return 0;
}
// Tichmotsovoimatran.cpp
#include <iostream.h>
#include <conio.h>
int a[10][10];
int b[10][10];
int m,n;
int i,j,c;
main()
```

```
{
    clrscr();
    cout<<"Ban nhap so dong cua ma tran m<10 "<<"\n";
    cin>>m;
    cout<<"Ban nhap so cot cua ma tran n<10 "<<"\n";
    cin>>n;
    cout<<"Nhap cac phan tu cho ma tran a "<<"\n";
    for (i=1; i<=m; i++)
        for (j=1; j<=n; j++)
          {
              cout<<"Ban nhap a["<<i<<"]["<<j<<"]= ";
              cin>>a[i][j];
          }
     cout<<"Ban nhap vao he so can nhan c=  ";
     cin>>c;
   // Tich cua mot so voi ma tran
     for (i=1; i<=m; i++)
        for (j=1; j<=n; j++)
          b[i][j]=a[i][j]*c;
        cout<<"He so c= "<<c<<"\n";
        cout<<"Ma tran a: "<<"\n";
        for (i=1; i<=m; i++)
            for (j=1; j<=n; j++)
              {
                  cout<<a[i][j]<<"   ";
                  if (j==n)
                    cout<<"\n";
              }
        cout<<"Tich cua "<<c<<"voi ma tran a la: "<<"\n";
        for (i=1; i<=m; i++)
          for (j=1; j<=n; j++)
                {
                  cout<<b[i][j]<<"   ";
                  if (j==n)
                    cout<<"\n";
                }
```

```
    getch();
    return 0;
}
```

◎ Bạn chạy thử chương trình tính tích của một số với ma trận.

◎ Bạn nhập vào số x=2 và nhập ma trận có 3 hàng, 4 cột như sau

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 3 | 4 | 5 | 4 |
| 4 | 6 | 3 | 2 |

Bạn sẽ có kết quả

| | | | |
|---|---|---|---|
| 2 | 4 | 6 | 8 |
| 6 | 8 | 10 | 8 |
| 8 | 12 | 6 | 4 |

◎ Bạn chạy thử chương trình tính bình phương của một ma trận.

◎ Nhập vào ma trận vuông 2 x 2 như sau:

| | |
|---|---|
| 3 | 2 |
| 1 | 4 |

◎ Bạn thấy kết quả của B mũ 2 là:

| | |
|---|---|
| 11 | 14 |
| 7 | 18 |

Chúng ta thử kiểm tra lại xem sao nhé!

$$B^2 = \begin{matrix} 3 & 2 \\ 1 & 4 \end{matrix} \times \begin{matrix} 3 & 2 \\ 1 & 4 \end{matrix} = \begin{matrix} 9+2 & 6+8 \\ 3+4 & 2+16 \end{matrix} = \begin{matrix} 11 & 14 \\ 7 & 18 \end{matrix}$$

◎ Bạn chạy chương trình tính lập phương của ma trận, bạn cũng nhập vào ma trận 2 x 2 như trên, bạn sẽ có kết quả như sau

| | |
|---|---|
| 47 | 78 |
| 39 | 86 |

Chúng ta thử kiểm tra lại nhé!

$$B^3 = \begin{matrix} 11 & 14 \\ 7 & 18 \end{matrix} \times \begin{matrix} 3 & 2 \\ 1 & 4 \end{matrix} = \begin{matrix} 33+14 & 22+56 \\ 21+18 & 14+72 \end{matrix} = \begin{matrix} 47 & 78 \\ 39 & 86 \end{matrix}$$

# ☎ Phần 7

## Bài tập sử dụng Cấu trúc (Struct) Hợp (Union) - Tập tin nhị phân - Lớp Lớp dẫn xuất và Phương thức ảo

### ✍ Dùng cấu trúc (Struct) nhập điểm HKI, HKII và cả năm của 3 học sinh, In nội dung ra màn hình

```
// Nhap_va_ThongBao_Ketqua_HocSinh.cpp
#include<iostream.h>
#include<conio.h>
char hoi;
main()
 {
    struct Hocsinh {
          char Hoten[40];
          int namsinh;
          float TBCMHKI;
          float TBCMHKII;
          float TBCMCN;
    };
    Hocsinh  HS1;
    Hocsinh  HS2;
    Hocsinh  HS3;
    cout<<"\n Nhap ho ten hoc sinh 1 (khong nhap khoang trang): ";
    cin>>HS1.Hoten;
    cout<<"\n Nam sinh: ";
```

```
cin>>HS1.namsinh;
cout<<"\n Trung binh cac mon Hoc ky I : ";
cin>>HS1.TBCMHKI;
cout<<"\n Trung binh cac mon Hoc ky II : ";
cin>>HS1.TBCMHKII;
cout<<"\n Trung binh cac mon ca nam : ";
cin>>HS1.TBCMCN;
cout<<"\n Nhap ho ten hoc sinh 2 (khong nhap khoang trang): ";
cin>>HS2.Hoten;
cout<<"\n Nam sinh: ";
cin>>HS2.namsinh;
cout<<"\n Trung binh cac mon Hoc ky I: ";
cin>>HS2.TBCMHKI;
cout<<"\n Trung binh cac mon Hoc ky II: ";
cin>>HS2.TBCMHKII;
cout<<"\n Trung binh cac mon ca nam: ";
cin>>HS2.TBCMCN;
cout<<"\n Nhap ho ten hoc sinh 3 (khong nhap khoang trang): ";
cin>>HS3.Hoten;
cout<<"\n Nam sinh: ";
cin>>HS3.namsinh;
cout<<"\n Trung binh cac mon Hoc ky I: ";
cin>>HS3.TBCMHKI;
cout<<"\n Trung binh cac mon Hoc ky II: ";
cin>>HS3.TBCMHKII;
cout<<"\n Trung binh cac mon ca nam: ";
cin>>HS3.TBCMCN;

cout<<"\n          KET QUA CUA HOC SINH";
cout<<"\n           Nam hoc 2001 - 2002";
cout<<"\n ========================================";
cout<<"\n Ho ten hoc sinh 1: "<<HS1.Hoten;
cout<<"\n Nam sinh: "<<HS1.namsinh;
cout<<"\n Trung binh cac mon hoc ky I: "<<HS1.TBCMHKI;
cout<<"\n Trung binh cac mon hoc ky II: "<<HS1.TBCMHKII;
cout<<"\n Trung binh cac mon ca nam: "<<HS1.TBCMCN;
```

```
    cout<<"\n Ho ten hoc sinh 2: "<<HS2.Hoten;
    cout<<"\n Nam sinh: "<<HS2.namsinh;
    cout<<"\n Trung binh cac mon hoc ky I: "<<HS2.TBCMHKI;
    cout<<"\n Trung binh cac mon hoc ky II: "<<HS2.TBCMHKII;
    cout<<"\n Trung binh cac mon ca nam: "<<HS2.TBCMCN;

    cout<<"\n Ho ten hoc sinh 3: "<<HS3.Hoten;
    cout<<"\n Nam sinh: "<<HS3.namsinh;
    cout<<"\n Trung binh cac mon hoc ky I: "<<HS3.TBCMHKI;
    cout<<"\n Trung binh cac mon hoc ky II: "<<HS3.TBCMHKII;
    cout<<"\n Trung binh cac mon ca nam: "<<HS3.TBCMCN;

    cout<<"\n ========================================";

    getch();
    return 0;
}
```

## Dùng hợp (Union), nhập điểm HKI, HKII và cả năm của 3 HS In nội dung ra màn hình

```
// Nhap_va_ThongBao_Ketqua_HocSinh.cpp
#include<iostream.h>
#include<conio.h>
char hoi;
main()
 {
    union Hocsinh {
          char Hoten[40];
          int namsinh;
          float TBCMHKI;
          float TBCMHKII;
          float TBCMCN;
```

```
} HS1,HS2,HS3;
cout<<"\n Nhap ho ten hoc sinh 1 (khong nhap khoang trang): ";
cin>>HS1.Hoten;
cout<<"\n Nam sinh: ";
cin>>HS1.namsinh;
cout<<"\n Trung binh cac mon Hoc ky I: ";
cin>>HS1.TBCMHKI;
cout<<"\n Trung binh cac mon Hoc ky II: ";
cin>>HS1.TBCMHKII;
cout<<"\n Trung binh cac mon ca nam: ";
cin>>HS1.TBCMCN;
cout<<"\n Nhap ho ten hoc sinh 2 (khong nhap khoang trang): ";
cin>>HS2.Hoten;
cout<<"\n Nam sinh: ";
cin>>HS2.namsinh;
cout<<"\n Trung binh cac mon Hoc ky I: ";
cin>>HS2.TBCMHKI;
cout<<"\n Trung binh cac mon Hoc ky II: ";
cin>>HS2.TBCMHKII;
cout<<"\n Trung binh cac mon ca nam: ";
cin>>HS2.TBCMCN;
cout<<"\n Nhap ho ten hoc sinh 3 (khong nhap khoang trang): ";
cin>>HS3.Hoten;
cout<<"\n Nam sinh: ";
cin>>HS3.namsinh;
cout<<"\n Trung binh cac mon Hoc ky I: ";
cin>>HS3.TBCMHKI;
cout<<"\n Trung binh cac mon Hoc ky II: ";
cin>>HS3.TBCMHKII;
cout<<"\n Trung binh cac mon ca nam: ";
cin>>HS3.TBCMCN;
cout<<"\n          KET QUA CUA HOC SINH";
cout<<"\n           Nam hoc 2001 - 2002";
cout<<"\n ========================================";
cout<<"\n Ho ten hoc sinh 1: "<<HS1.Hoten;
cout<<"\n Nam sinh: "<<HS1.namsinh;
```

```
    cout<<"\n Trung binh cac mon hoc ky I: "<<HS1.TBCMHKI;
    cout<<"\n Trung binh cac mon hoc ky II: "<<HS1.TBCMHKII;
    cout<<"\n Trung binh cac mon ca nam: "<<HS1.TBCMCN;

    cout<<"\n Ho ten hoc sinh 2: "<<HS2.Hoten;
    cout<<"\n Nam sinh: "<<HS2.namsinh;
    cout<<"\n Trung binh cac mon hoc ky I: "<<HS2.TBCMHKI;
    cout<<"\n Trung binh cac mon hoc ky II: "<<HS2.TBCMHKII;
    cout<<"\n Trung binh cac mon ca nam: "<<HS2.TBCMCN;
    cout<<"\n Ho ten hoc sinh 3: "<<HS3.Hoten;
    cout<<"\n Nam sinh: "<<HS3.namsinh;
    cout<<"\n Trung binh cac mon hoc ky I: "<<HS3.TBCMHKI;
    cout<<"\n Trung binh cac mon hoc ky II: "<<HS3.TBCMHKII;
    cout<<"\n Trung binh cac mon ca nam: "<<HS3.TBCMCN;

    cout<<"\n ========================================";

    getch();
    return 0;
}
```

## ✍ Nhập và đọc một tập tin dạng nhị phân có phần mở rộng .DAT gồm họ tên, năm sinh, TBCMHKI, TBCMHKII và TBCMCN của các học sinh

```
//Taptin_Nhiphan_Viet.cpp
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
```

```
{
  clrscr();
  struct Hocsinh
          {
            char Hoten[40];
            int  Namsinh;
            float TBCMHKI;
            float TBCMHKII;
            float TBCMCN;
          }HS;
  FILE *taptin;
  if ((taptin=fopen("c:\\HOCSINH.DAT","wb"))==NULL)
      {
        cout<<"Khong the mo tap tin HOCSINH.DAT";
        exit(1);
      }
  do
   {
       cout<<"\n Ho va ten cua Hoc sinh: ";
       cin>>HS.Hoten;
       cout<<"\n Nam sinh: ";
       cin>>HS.Namsinh;
       cout<<"\n Trung binh cac mon hoc ky I: ";
       cin>>HS.TBCMHKI;
       cout<<"\n Trung binh cac mon hoc ky II: ";
       cin>>HS.TBCMHKII;
       cout<<"\n Trung binh cac mon ca nam: ";
       cin>>HS.TBCMCN;

       cout<<"\n Co nhap them giao vien khong (c/k)?";
   }
   while (getche()=='c');
  fclose(taptin);
  return 0;
}
```

Bạn chạy chương trình này và nhập nội dung cho các học sinh. Sau đó chạy chương trình sau để đọc.

```
//Taptin_Nhiphan_Doc.cpp
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
 {
    clrscr();
    struct Hocsinh
            {
              char Hoten[40];
              int  Namsinh;
              float TBCMHKI;
              float TBCMHKII;
              float TBCMCN;
            }HS;
    FILE *taptin;
    if ((taptin = fopen("c:\\HOCSINH.DAT","rb"))==NULL)
        {
          cout<<"Khong the mo tap tin HOCSINH.DAT";
          exit(1);
        }
    while (fread(&HS,sizeof(HS),1,taptin)==1)
      {
          cout<<"\n Ho va ten cua hoc sinh: "<<HS.Hoten;
          cout<<"\n Nam sinh: "<<HS.Namsinh;
          cout<<"\n Trung binh cac mon hoc ky I: "<<HS.TBCMHKI;
          cout<<"\n Trung binh cac mon hoc ky II: "<<HS.TBCMHKII;
          cout<<"\n Trung binh cac mon ca nam: "<<HS.TBCMCN;
      }
    fclose(taptin);
    getch();
    return 0;
 }
```

# ✍ Chương trình mẫu dùng lớp

⇨ Bạn hãy tạo một chương trình trong đó có xây dựng một lớp có tên là ngonngu. Trong lớp này có các phương thức thành phần là Pascal, C, VisualBasic, VisualC, Java.

⇨ Trong các phương thức này là những ghi chú về ngôn ngữ, bạn thiết kế sao cho khi cần xem ngôn ngữ nào thì bạn sẽ thấy được các thông tin về ngôn ngữ đó.

```
// DUNGLOP
#include<iostream.h>
#include<conio.h>
class ngonngu
 {
   public :
   void Pascal();
   void C();
   void VisualC();
   void VisualBasic();
   void Java();
 };
 void ngonngu :: Pascal(void)
    {
        cout<<" Ban muon biet gi ve ngon ngu PASCAL";
        cout<<"\n Day la mot ngon ngu lap trinh cau truc";
        cout<<"\n Va no con duoc xem la mot ngon ngu day hoc";
        cout<<"\n No duoc su dung rong rai trong nhieu doi tuong";
        cout<<"\n Nhat la doi voi hoc sinh pho thong";
        cout<<"\n Va cac sinh vien nam dau cua cac bac cao dang";
        cout<<"\n Va Dai hoc";
        cout<<"\n Chac cac ban cung biet nhieu ve ngon ngu nay";
    };
void ngonngu :: C(void)
    {
        cout<<" Ban muon biet gi ve ngon ngu C";
        cout<<"\n Day la mot ngon ngu lap trinh cau truc";
        cout<<"\n Va no duoc dung trong nhieu linh vuc";
```

```
        cout<<"\n No duoc su dung rong rai trong nhieu doi tuong";
        cout<<"\n Nhat la doi voi hoc sinh pho thong";
        cout<<"\n Va cac sinh vien nam dau cua cac bac cao dang";
        cout<<"\n Va Dai hoc";
        cout<<"\n Chac cac ban cung biet nhieu ve ngon ngu nay";
        cout<<"\n No manh hon ngon ngu Pascal";
        cout<<"\n Cac sinh vien deu biet ngon ngu nay";
    };
void ngonngu :: VisualC(void)
    {
        cout<<" Ban muon biet gi ve ngon ngu VISUAL C++";
        cout<<"\n Day la mot ngon ngu lap trinh huong doi tuong";
        cout<<"\n Va no duoc dung trong nhieu linh vuc";
        cout<<"\n No duoc su dung rong rai trong nhieu doi tuong";
        cout<<"\n Nhat la doi voi nguoi lam khoa hoc";
        cout<<"\n Va cac sinh vien  cua cac bac cao dang";
        cout<<"\n Va Dai hoc";
        cout<<"\n Chac cac ban cung biet nhieu ve
                        ngon ngu nay";
        cout<<"\n No rat manh va ung dung tot hon ngon ngu C";
        cout<<"\n Lap trinh hoi kho, nhieu khai niem moi";
    };
void ngonngu :: VisualBasic(void)
    {
        cout<<" Ban muon biet gi ve ngon ngu VISUAL BASIC ";
        cout<<"\n Day la mot ngon ngu lap trinh huong doi tuong ";
        cout<<"\n Va no duoc dung trong nhieu linh vuc";
        cout<<"\n No duoc su dung rong rai trong nhieu doi tuong";
        cout<<"\n Nhat la doi voi nguoi lap trinh";
        cout<<"\n No chua nam trong chuong trinh cua bac cao dang";
        cout<<"\n Va Dai hoc";
        cout<<"\n Cac ban da su dung ngon ngu nay chua";
        cout<<"\n No rat manh va Form chuong trinh co hinh dang dep";
        cout<<"\n Lap trinh de hon VisualC++";
        cout<<"\n Cac cong ty thuong tuyen nguoi biet ngon ngu nay";
    };
```

```
void ngonngu :: Java(void)
    {
        cout<<" Ban muon biet gi ve ngon ngu JAVA";
        cout<<"\n Day la mot ngon ngu lap trinh huong doi tuong";
        cout<<"\n Va no duoc dung trong nhieu linh vuc";
        cout<<"\n No duoc su dung rong rai trong nhieu doi tuong";
        cout<<"\n Nhat la doi voi nguoi lam khoa hoc, thiet ke trang WEB";
        cout<<"\n Va cac sinh vien ´cua cac bac cao dang";
        cout<<"\n Va Dai hoc";
        cout<<"\n Chac cac ban cung biet nhieu ve ngon ngu nay";
        cout<<"\n No rat manh va ung dung tot trong nhieu linh vuc";
        cout<<"\n Lap trinh hoi kho, nhieu khai niem moi";
    };

    ngonngu Xem;
main()
 {
   int banchon;
   do
    {
     cout << "\n Ban chon 1, 2, 3, 4, 5 hoac 6? "
          << "\n 1- Xem ngon ngu PASCAL"
          << "\n 2- Xem ngon ngu C"
          << "\n 3- Xem ngon ngu Visual C++"
          << "\n 4- Xem ngon ngu Visual Basic"
          << "\n 5- Xem ngon ngu JAVA"
          << "\n 6- Thoat khoi chuong trinh";
     cin >> banchon;
     switch(banchon)
         {
          case 1:
            {
                  clrscr();
                  Xem.Pascal();
                  break;
            }
```

```
        case 2 :
          {
                clrscr();
                Xem.C();
                break;
          }
        case 3 :
          {
                clrscr();
                Xem.VisualC();
                break;
          }
        case 4 :
          {
                clrscr();
                Xem.VisualBasic();
                break;
          }
        case 5:
          {
                clrscr();
                Xem.Java();
                break;
          }
        case 6 : break;
        default : cout<<"Sai roi chon 1, 2, 3, 4, 5 hoac 6";
      }
  } while(banchon!=6);
  return 0;
}
```

⇨ Trong chương trình trên, bạn cài đặt một lớp có tên la ngonngu.

⇨ Trong lớp này, bạn cài đặt các phương thức thành phần như yêu cầu của đầu bài.

⇨ Các phương thức được khai báo thông qua câu lệnh

Lớp cơ sở :: Phương thức thành phần ();

Trong chương trình trên là các câu khai báo sau

```
ngonngu :: Pascal();
ngonngu :: C();
ngonngu :: VisualC();
ngonngu :: VisualBasic();
ngonngu :: Java();
```

⇨ Trong các phương thức thành phần này là những câu lệnh xuất ra màn hình chú thích về ngôn ngữ .

⇨ Vòng lặp do .....while dùng để xuất hiện Menu của chương trình cho người sử dụng chọn lựa.

⇨ Biến Banchon co kiểu nguyên dùng trong phát biểu Switch để thực hiện khi người sử dụng chọn một số tương ứng từ 1 đến 6

⇨ Qua chương trình này, các bạn sẽ thấy rõ hơn về Lớp

◎ Khi chạy thử chương trình, bạn sẽ thấy xuất hiện một Menu như sau :

```
Ban chon 1, 2, 3, 4, 5 hoac 6?
1- Xem ngon ngu PASCAL
2- Xem ngon ngu C
3- Xem ngon ngu Visual C++
4- Xem ngon ngu Visual Basic
5- Xem ngon ngu JAVA
6- Thoat khoi chuong trinh
```

◎ Bạn gõ vào số cần xem, giả sử bạn gõ số 3, bạn sẽ có kết quả như sau:

```
Ban muon biet gi ve ngon ngu VISUAL C++
Day la mot ngon ngu lap trinh huong doi tuong
Va no duoc dung trong nhieu linh vuc
No duoc su dung rong rai trong nhieu doi tuong
Nhat la doi voi nguoi lam khoa hoc
Va cac sinh vien  cua cac bac cao dang
Va Dai hoc
Chac cac ban cung biet nhieu ve ngon ngu nay
No rat manh va ung dung tot hon ngon ngu C
Lap trinh hoi kho, nhieu khai niem moi
```

Ban chon 1, 2, 3, 4, 5 hoac 6?
1- Xem ngon ngu PASCAL
2- Xem ngon ngu C
3- Xem ngon ngu Visual C++
4- Xem ngon ngu Visual Basic
5- Xem ngon ngu JAVA
6- Thoat khoi chuong trinh

◎ Nếu bạn chọn số khác, giả sử số 5, bạn sẽ thấy như sau

```
Ban muon biet gi ve ngon ngu JAVA
Day la mot ngon ngu lap trinh huong doi tuong
Va no duoc dung trong nhieu linh vuc
No duoc su dung rong rai trong nhieu doi tuong
Nhat la doi voi nguoi lam khoa hoc, thiet ke trang WEB
Va cac sinh vien  cua cac bac cao dang
Va Dai hoc
Chac cac ban cung biet nhieu ve ngon ngu nay
No rat manh va ung dung tot trong nhieu linh vuc
Lap trinh hoi kho, nhieu khai niem moi
        Ban chon 1, 2, 3, 4, 5 hoac 6 ?
          1- Xem ngon ngu PASCAL
          2- Xem ngon ngu C
          3- Xem ngon ngu Visual C++
          4- Xem ngon ngu Visual Basic
          5- Xem ngon ngu JAVA
          6- Thoat khoi chuong trinh
```

Bạn thử chạy thử và xem các trường hợp còn lại.

# ✍ Chương trình minh hoạ
# Lớp dẫn xuất từ một lớp cơ sở

```
#include<iostream.h>
#include<conio.h>
class Vatly
  {
    public :
      void lopbay(void);
```

```
      void loptam(void);
  };
class Vatlymot : public Vatly
  {
    public :
      void lopchin(void);
  }ketqua;
  void Vatly :: lopbay(void)
    {
        cout<<"\n Ban se hoc nhung kien thuc ve co hoc";
    }
  void Vatly :: loptam(void)
    {
        cout<<"\n Ban se hoc nhung kien thuc ve nhiet hoc";
    }
  void Vatlymot :: lopchin(void)
    {
        cout<<"\n Ban se hoc nhung kien thuc ve dien hoc";
    }
  main()
   {
     clrscr();
     cout<<"\n Lop bay "; ketqua.lopbay();
     cout<<"\n Lop tam " ; ketqua.loptam();
     cout<<"\n Lop chin "; ketqua.lopchin();
     getch();
     return 0;
   };
```

⇨ Chương trình này rất là đơn giản, chúng tôi giới thiệu như là một cái sườn, các bạn nắm vững thì sau này sẽ lập trình một cách dễ dàng khi dùng đến lớp và lớp dẫn xuất.

⇨ Ở chương trình trên, đầu tiên chúng tôi cài đặt một lớp cơ sở có tên là Vatly.

⇨ Trong lớp Vatly này, chúng tôi cài hai phương thức thành phần, đó là lopbay() và loptam().

⇨ Kế đến chúng tôi cài một lớp dẫn xuất có tên Vatlymot, lớp này thừa kế lớp cơ sở Vatly.

⇨ Trong lớp này, chúng tôi có một phương thức đó là lopchin().

⇨ Đối tượng ketqua có kiểu Vatlymot.

⇨ Bên dưới là nội dung của các phương thức.

⇨ Vậy chương trình trên là tạo một lớp dẫn xuất có tên là Vatlymot thừa kế lớp cơ sở có tên Vatly.

⇨ Nếu chú ý trong hàm main(), bạn thấy chúng tôi dùng đối tượng có tên ketqua để tác động đến tất cả các phương thức thành phần như lopbay(), loptam() và lopchin(), đối tượng ketqua có kiểu Vatlymot.

⇨ Sở dĩ được là do lớp Vatlymot dẫn xuất từ lớp cơ sở Vatly.

◎ Khi chạy thử, bạn sẽ có kết quả hiện ra trên màn hình như sau

```
Ban se hoc nhung kien thuc ve co hoc
Ban se hoc nhung kien thuc ve nhiet hoc
Ban se hoc nhung kien thuc ve dien hoc
```

# Chương trình minh hoạ Lớp dẫn xuất từ nhiều lớp cơ sở

Trong chương trình trên, chúng tôi dùng một lớp dẫn xuất từ một lớp cơ sở, trong chương trình này, chúng tôi dùng một lớp dẫn xuất từ hai lớp cơ sở.

Bạn xem code của chương trình như sau:

```
#include<iostream.h>
#include<conio.h>
class Vatly
 {
   public :
      void lopbay(void);
      void loptam(void);
 };
class Vatlymot
 {
   public :
```

```
      void lopchin(void);
  };
class Vatlyhai : public Vatly, public Vatlymot
  {
    public :
      void lopmuoi(void);
  }ketqua;
  void Vatly :: lopbay(void)
    {
        cout<<"\n Ban se hoc nhung kien thuc ve co hoc";
    }
  void Vatly :: loptam(void)
    {
        cout<<"\n Ban se hoc nhung kien thuc ve nhiet hoc";
    }
  void Vatlymot :: lopchin(void)
    {
        cout<<"\n Ban se hoc nhung kien thuc ve dien hoc";
    }
  void Vatlyhai :: lopmuoi(void)
   {
      cout<<"\n Ban hoc nhung kien thuc ve co hoc sau hon";
   }
  main()
   {
     clrscr();
     cout<<"\n Lop bay"; ketqua.lopbay();
     cout<<" \n Lop tam" ; ketqua.loptam();
     cout<<"\n Lop chin"; ketqua.lopchin();
     cout<<"\n Lop muoi"; ketqua.lopmuoi();
     getch();
     return 0;
  };
```

⇨ Bạn thấy trong chương trình trên, đầu tiên chúng tôi cài đặt hai lớp cơ sở, lớp thứ nhất có tên là Vatly, lớp thứ hai có tên là Vatlymot.

⇨ Trong lớp cơ sở Vatly có hai phương thức thành phần, đó là lopbay() và loptam().

⇨ Trong lớp cơ sở Vatlymot có một phương thức đó là lopchin().

⇨ Một lớp dẫn xuất từ hai lớp cơ sở trên, có tên là Vatlyhai. Trong lớp dẫn xuất này, chúng tôi cài thêm một phương thức thành phần có tên là lopmuoi().

⇨ Đối tượng ketqua có kiểu là VatlyHai.

⇨ Bên dưới là nội dung cài đặt trong các phương thức thành phần.

⇨ Trong hàm main(), đối tượng ketqua đã tác động đến tất cả các phương thức như lopbay(), loptam() , lopchin() và lopmuoi(). Điều này bạn đã được giải thích ở trên.

◎ Khi chạy thử chương trình, bạn sẽ có kết quả như sau:

```
Ban se hoc nhung kien thuc ve co hoc
Ban se hoc nhung kien thuc ve nhiet hoc
Ban se hoc nhung kien thuc ve dien hoc
Ban hoc nhung kien thuc ve co hoc sau hon
```

# ✍ Chương trình minh hoạ Lớp vừa dẫn xuất từ một lớp cơ sở của một lớp dẫn xuất khác

```
#include<iostream.h>
#include<conio.h>
class Vatly
  {
    public :
       void lopbay(void);
       void loptam(void);
  };
class Vatlymot
  {
     public :
       void lopchin(void);
  };
```

```
class Vatlyhai : public Vatly, public Vatlymot
 {
    public :
      void lopmuoi(void);
 };
class Vatlyba : public Vatlyhai
 {
    public :
         void lopmuoimot(void);
         void lopmuoihai(void);
 }ketqua;
 void Vatly :: lopbay(void)
   {
        cout<<"\n Ban se hoc nhung kien thuc ve co hoc";
   }
 void Vatly :: loptam(void)
   {
        cout<<"\n Ban se hoc nhung kien thuc ve nhiet hoc";
   }
 void Vatlymot :: lopchin(void)
   {
        cout<<"\n Ban se hoc nhung kien thuc ve dien hoc";
   }
 void Vatlyhai :: lopmuoi(void)
  {
     cout<<"\n Ban hoc nhung kien thuc ve co hoc sau hon";
  }
 void Vatlyba :: lopmuoimot(void)
  {
     cout<<"\n Ban hoc nhung kien thuc ve nhiet va dien sau hon";
  }
 void Vatlyba :: lopmuoihai(void)
  {
     cout<<"\n Ban hoc nhung kien thuc ve dao dong, dien va nguyen tu";
  }
 main()
```

```
{
  clrscr();
  cout<<"\n Lop bay"; ketqua.lopbay();
  cout<<" \n Lop tam" ; ketqua.loptam();
  cout<<"\n Lop chin"; ketqua.lopchin();
  cout<<"\n Lop muoi"; ketqua.lopmuoi();
  cout<<"\n Lop muoi mot"; ketqua.lopmuoimot();
  cout<<"\n Lop muoihai"; ketqua.lopmuoihai();
  getch();
  return 0;
};
```

⇨ Đầu tiên chúng ta cài đặt hai lớp cơ sở có tên là Vatly và Vatlymot.

⇨ Trong lớp cơ sở có tên Vatly có hai phương thức thành phần, đó là lopbay() và loptam().

⇨ Trong lớp cơ sở có tên Vatlymot có một phương thức thành phần, đó là lopchin().

⇨ Tiếp đến chúng ta tạo một lớp dẫn xuất có tên Vatlyhai thừa kế hai lớp cơ sở trên, đó là lớp Vatly và lớp Vatlymot qua câu lệnh

```
class Vatlyhai : public Vatly, public Vatlymot
```

⇨ Trong lớp dẫn xuất này, chúng tôi cài đặt một phương thức có tên là lopmuoi().

⇨ Kế đến chúng tôi cài đặt thêm một lớp dẫn xuất có tên Vatlyba thừa kế lớp có tên Vatlyhai qua câu lệnh sau

```
class Vatlyba : public Vatlyhai
```

⇨ Trong lớp dẫn xuất này, chúng tôi cài thêm hai phương thức, đó là lopmuoimot() và lopmuoihai().

⇨ Tạo một đối tượng có tên là ketqua có kiểu là Vatlyba, bạn thấy qua đoạn code sau

```
class Vatlyba : public Vatlyhai
 {
   public :
       void lopmuoimot(void);
         void lopmuoihai(void);
```

}ketqua;

⇨ Vậy đến đây, bạn sẽ thấy lớp Vatlyba sẽ có đầy đủ các phương thức thành phần của các lớp, sở dĩ có được điều này là do khái niệm về sự thừa kế.

⇨ Tiếp đến là chúng tôi viết nội dung của các phương thức thành phần của các lớp.

⇨ Trong hàm main() các bạn thấy truy xuất nội dung của các phương thức thông qua đối tượng kết quả.

◎ Chạy thử chương trình, bạn sẽ thấy kết quả như sau

```
Ban se hoc nhung kien thuc ve co hoc
Ban se hoc nhung kien thuc ve nhiet hoc
Ban se hoc nhung kien thuc ve dien hoc
Ban hoc nhung kien thuc ve co hoc sau hon
Ban hoc nhung kien thuc ve nhiet va dien sau hon
Ban hoc nhung kien thuc ve dao dong, dien va nguyen tu
```

# Dùng phương thức ảo

```
#include<iostream.h>
#include<conio.h>
 class Bia
  {
   public :
     virtual void Chetaobaobi(void);
  };
 class Heineken:public Bia
 {
   public :
      void Chetaobaobi(void);
 }Hinhdang1;
  void Heineken:: Chetaobaobi(void)
      {
           cout<<"\n Lam bang nhom ";
           cout<<"\n Hinh khoi tron, co chieu cao la 50cm";
           cout<<"\n Son mau xanh, chu trang";
      }
```

```
class Tiger : public Bia
   {
        public :
           void Chetaobaobi(void);
   }Hinhdang2;
void Tiger :: Chetaobaobi(void)
  {
        cout<<"\n Lam bang nhom";
        cout<<"\n Hinh khoi tron, co chieu cao la 65 cm";
        cout<<"\n Son mau xanh duong, chu vang";
  }
class Bababa : public Bia
   {
        public :
           void Chetaobaobi(void);
   }Hinhdang3;
void Bababa :: Chetaobaobi(void)
  {
        cout<<"\n Lam bang nhom";
        cout<<"\n Hinh khoi tron, co chieu cao la 50 cm";
        cout<<"\n Son mau trang, chu do";
  }
class Saigonxanh : public Bia
   {
        public :
           void Chetaobaobi(void);
   }Hinhdang4;
void Saigonxanh :: Chetaobaobi(void)
  {
        cout<<"\n  Lam bang thuy tinh";
        cout << "\n Hinh cai chai, co chieu cao la 65 cm";
        cout<< "\n Chai mau nau, co duong kinh lon";
  }
class Saigondo : public Bia
   {
        public :
```

```
            void Chetaobaobi(void);
      }Hinhdang5;
   void Saigondo :: Chetaobaobi(void)
     {
          cout<<"\n Lam bang thuy tinh";
          cout<<"\n Hinh cai chai, co chieu cao la 65cm";
          cout<<"\n Chai mau nau, co duong kinh nho";
     }
   class Tigerchainho : public Bia
     {
          public :
             void Chetaobaobi(void);
     }Hinhdang6;
   void Tigerchainho :: Chetaobaobi(void)
     {
          cout<<"\n Lam bang thuy tinh";
          cout<<"\n Hinh cai chai, co chieu cao la 80cm";
          cout<<"\n Chai mau nau, co duong kinh nho";
     }
main()
  {
    int banchon;
    do
     {
      cout<<"\n Ban chon 1, 2 , 3, 4, 5, 6 hoac 7?"
            << "\n 1- Che tao lon bia Heineken"
            << "\n 2- Che tao lon bia Tiger"
            << "\n 3- Che tao bia lon 333"
            << "\n 4- Che tao bia chai Saigon xanh"
            << "\n 5- Che tao bia chai Saigon do"
            << "\n 6- Che tao bia chai Tiger nho"
            << "\n 7- Thoat khoi chuong trinh";
     cin>>banchon;
     switch(banchon)
         {
           case 1:
```

```
  {
      clrscr();
      Hinhdang1.Chetaobaobi();
        break;
  }
case 2 :
  {
        clrscr();
        Hinhdang2.Chetaobaobi();
        break;
  }
case 3 :
  {
        clrscr();
        Hinhdang3.Chetaobaobi();
        break;
  }
 case 4:
  {
       clrscr();
       Hinhdang4.Chetaobaobi();
       break;
  }
 case 5 :
  {
       clrscr();
       Hinhdang5.Chetaobaobi();
        break;
  }
 case 6 :
  {
       clrscr();
       Hinhdang6.Chetaobaobi();
        break;
  }
 case 7 : break;
```

```
            default : cout<< " Sai roi chon 1, 2, 3, 4, 5, 6 hoac 7 ";
        }
    } while(banchon !=7);
    return 0;
}
```

⇨ Trong chương trình này, bạn sẽ tạo một phương thức ảo có tên là Chetaobaobi() trong lớp cơ sở Bia, bạn khai báo qua câu lệnh như sau virtual Chetaobaobi();

⇨ Kế đến là bạn cài đặt lớp dẫn xuất Heineken thừa kế lớp cơ sở Bia.

⇨ Trong phương thức này, bạn cài đặt một phương thức có tên Chetaobaobi(). Tên này giống với tên của phương thức ảo trong lớp Bia, chỉ khác là không có từ virtual ở phía trước.

⇨ Tiếp theo bạn sẽ xác lập nội dung của phương thức.

⇨ Tương tự như vậy với các lớp tiếp theo. Bạn có thể thấy ở chương trình trên.

⇨ Trong chương trình này chúng ta chỉ cần gửi một thông điệp Chetaobaobi đến các đối tượng khác nhau, chúng đều thực hiện.

◎ Chạy thử chương trình, bạn sẽ có kết quả như sau

◎ Đầu tiên sẽ xuất hiện một Menu

```
Ban chon 1, 2 , 3, 4, 5, 6 hoac 7?
 1- Che tao lon bia Heineken
 2- Che tao lon bia Tiger
 3- Che tao bia lon 333
 4- Che tao bia chai Saigon xanh
 5- Che tao bia chai Saigon do
 6- Che tao bia chai Tiger nho
 7- Thoat khoi chuong trinh
```

◎ Bạn chọn số tương ứng để có kết quả chế tạo bao bì của các loại bia.

**BẠN SẼ THÀNH THẠO C++ CHỈ TRONG 21 NGÀY**

**ĐẬU QUANG TUẤN**

*Chịu trách nhiệm xuất bản :* TRẦN ĐÌNH VIỆT

*Biên tập* : TRUNG HIẾU

*Sửa bản in* : HƯƠNG HUÊ - NGUYỄN BẢO THÙY

*Trình bày* : ĐẬU THỤY QUỲNH TRÂM

*Bìa* : THIÊN Ý - ĐẬU MINH NHẬT

**NHÀ XUẤT BẢN TỔNG HỢP THÀNH PHỐ HỒ CHÍ MINH**

62 Nguyễn Thị Minh Khai - Q.1

***Điện thoại:*** *8225340 - 8296764 - 8220405 - 8296713 - 8223637*

***Fax:*** *84.8.8222726* • ***Email:*** *nxbtphcm@bdvn.vnd.net*

*Thực hiện liên doanh:* **ĐẬU QUANG TUẤN**

---

In 1.000 cuốn, khổ 14,5 x 20,5 cm. Tại **Công Ty Cổ Phần IN KHÁNH HỘI** - 27 Hoàng Diệu, P.12, Q.4, Tp. Hồ Chí Minh. GPXB số: 399-222/XB-QLXB ngày 11.4.2003. In xong và nộp lưu chiểu tháng 10/2004.

# SÁCH PHÁT HÀNH NĂM 2005


Tất cả những tập sách trên được phát hành