CHƯƠNG 3: CẤU TRÚC ĐIỀU KHIỂN

LỆNH

- Một chương trình đang chạy nghĩa là nó đang thực thi các câu ***lệnh***.
- Thứ tự mà các câu lệnh được thực hiện được gọi là **dòng điều khiển** (flow control)**.**
- Các câu lệnh đang thực thi có ***sự điều khiển*** của CPU, khi CPU hoàn thành sẽ được chuyển giao tới một lệnh khác.
- Đặc trưng dòng điều khiển trong một chương trình là tuần tự.

LỆNH

- Dòng điều khiển xem xét và quyết định lệnh nào được thực thi và lệnh nào không được thực thi trong quá trình chạy, vì thế làm ảnh hưởng đến kết quả toàn bộ của chương trình
- C++ cung cấp những hình thức lệnh khác nhau cho các mục đích khác nhau.

LỆNH

- ***Lệnh khai báo***: sử dụng để định nghĩa các biến
- ***Lệnh gán*** : sử dụng để tính toán đại số đơn giản
- ***Lệnh rẽ nhánh*** được sử dụng để chỉ định việc thực thi các lệnh, phụ thuộc vào kết quả của một điều kiện luận lý (True, False)
- ***Lệnh lặp*** được sử dụng để chỉ định các tính toán cần được lặp cho tới khi một điều kiện luận lý nào đó được thỏa.

LỆNH ĐƠN & LỆNH PHỨC

- **Lệnh đơn** là một sự tính toán được kết thúc bằng dấu **;**
- Nhiều lệnh đơn có thể kết nối lại thành một **lệnh phức** bằng cách rào chúng bên trong các dấu **{ }**

Ví dụ:

```
{
  int min, i = 10, j = 20;
  min = (i < j ? i : j);
  min + 5;
  cout << min << '\n';
  ;
}
```

Lệnh rỗng

Lệnh vô dụng

CẤU TRÚC RẼ NHÁNH

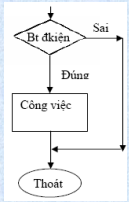
Cấu trúc điều kiện: *if* và *else*

- **Dạng 1:**

  Cú pháp:

  if (*biểu thức*)

  ***lệnh*;**



- Nếu *biểu thức được ước lượng với kết* quả khác 0 (đúng) thì *lệnh* được thực thi.
- Ngược lại, không làm gì cả.

Cấu trúc điều kiện: *if* và *else*

Ví dụ:

```
#include <iostream.h>
#include <conio.h>
void main ()
{
        float a;
        cout<<"Nhap a = ";
        cin>>a;
        if (a !=0 )
        cout<<"Nghich dao cua "<<a<<"la "<<1/a;
        getch();
}
```

Cấu trúc điều kiện: *if* và *else*

Cú pháp:

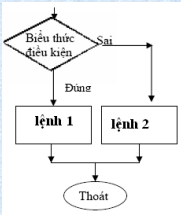
if (<*Biểu thức điều kiện*>**)**

<lệnh 1>

else

<lệnh 2>

Lưu đồ cú pháp:



Cấu trúc điều kiện: *if* và *else*

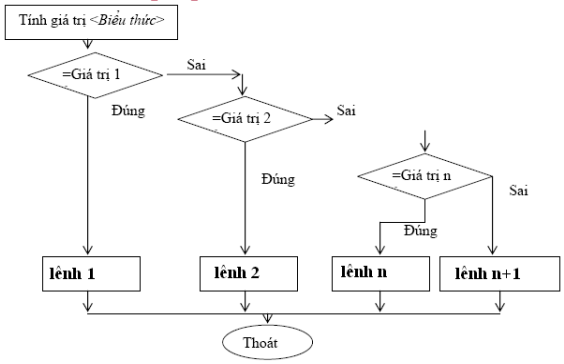
Ví dụ:

```
#include <iostream.h>
#include <conio.h>
void main ()
{
    float a;
    cout<<"Nhap a = ";
    cin>>a;
    if (a !=0 )
        cout<<"Nghich dao cua "<<a<<"la"<<1/a;
    else
        cout<<"Khong the tim duoc nghich dao cua a";
    getch();
}
```

Cấu trúc lựa chọn (switch)

```
switch (<Biểu thức giá trị>)
{
case giá trị 1:
    Khối lệnh thực hiện công việc 1;
    break;
…
case giá trị n:
    Khối lệnh thực hiện công việc n;
    break;
[default :
    Khối lệnh thực hiện công việc mặc định;
    break;]
}
```

Lưu đồ cú pháp



Ví dụ:

```
#include <iostream.h>
#include<conio.h>
void  main ()
{
      int n, phandu;
      cout<<" Nhap vao so nguyen : "; cin>>n;
      phandu=songuyen % 2;
      switch(phandu)
      {
        case 0: cout<<n<<" la so chan ";
                break;
        case 1: cout<<n<<" la so le ";
                break;
      }
```

BÀI TẬP Phần If

Bái 1:

Nhập 1 số n>=0. Tính và xuất căn bậc hai của n.

HD: dùng hàm **sqrt(a)=**

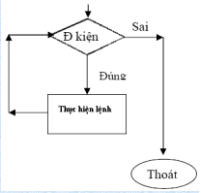
CẤU TRÚC LẶP

Vòng lặp while

Cú pháp

while (<*biểu thức điều kiện*>)
***lệnh*;**

Lưu đồ cú pháp



Vòng lặp while

Ví dụ:

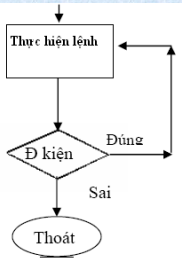
```
#include <iostream.h>
#include<conio.h>
int main ()
{
   int i, n, sum;
   i = 1;
   sum = 0;
   while (i <= n)
   {
     sum += i;
     i++;
   }
}
```

Vòng lặp do… while

Cú pháp:

do
<Lệnh>
while (<*Biểu thức điều kiện*>**);**

Lưu đồ cú pháp:



Ví dụ:

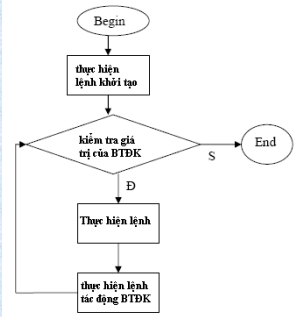
```
#include <iostream.h>
#include<conio.h>
void main ()
{
    int i;
    clrscr();
    cout<<"Day so tu 1 den 10 :";
    i=1;
    do
    {
        cout<<setw(3)<<i;
        i+=1;
    } while (i<=10);
getch(); }
```

Vòng lặp for

Cú pháp:

for (*biểu thức khởi tạo*; *biểu thứcđiều kiện*; *biểu thức điều khiển lặp*)
    <lệnh>

Lưu đồ cú pháp



Ví dụ:tính tổng n số nguyên

```
#include <iostream.h>
#include<conio.h>
void main ()
{
        int i, n, sum;
        cout<<"Nhap so nguyen n : "; cin>>n;
        sum = 0;
        for (i = 1; i <= n; i++)
          sum += i;
        cout<<" tong cua "<<n<<"so nguyen la : "<<sum;
        getch();
 }
```

Lưu ý:

- ✓C++ cho phép biểu thức đầu tiên trong vòng lặp for là một định nghĩa biến. Ví dụ:

```
for (int i = 1; i <= n; ++i)
    sum += i;
```

- ✓Bất kỳ biểu thức nào trong 3 biểu thức của vòng lặp for đều có thể rỗng. Ví dụ:

```
for (; i != 0;) // tương đương với: while (i != 0)
```

- ✓Xóa tất cả các biểu thức cho chúng ta một vòng lặp vô hạn

```
for (;;) // vòng lặp vô hạn
```

Các lệnh rẽ nhánh và lệnh nhảy

Lệnh continue

- ***Lệnh continue***: dừng lần lặp hiện tại của một vòng lặp và nhảy tới lần lặp kế tiếp.
- ***Lệnh continue:*** áp dụng tức thì cho vòng lặp gần với lệnh continue. Không sử dụng lệnh continue bên ngoài vòng lặp.

Lệnh continue

- Trong vòng lặp ***while*** và vòng lặp ***do-while***, chương trình sẽ nhảy đến kiểm tra điều kiện lặp **expression** ngay lập tức khi gặp lệnh **continue.** Lần lặp tiếp theo sẽ được thực thi khi điều kiện để lặp vẫn cho phép.

  while (expression)

  hoặc

  do……while (expression)

Lệnh continue

- Trong vòng lặp ***for***, lần lặp kế tiếp khởi đầu từ biểu thức thứ ba của vòng lặp.

for (*biểu thức khởi tạo*; *biểu thứcđiều kiện*; *biểu thức điều khiển lặp*)

- Sau đó kiểm tra biểu thức điều kiện, và chỉ lặp lại công việc nếu biểu thức điều kiện thỏa.

Ví dụ: *một vòng lặp thực hiện đọc một số, xử lý nó nhưng bỏ qua những số âm, và dừng khi số là 0, có thể diễn giải như sau*:

```
do
{
    cin >> num;
    if (num < 0) continue;
          cout<<num;
} while (num != 0);
```

Điều này tương đương với:

```
do
{
    cin >> num;
    if (num >= 0)
    {
          cout<<num;
    }
} while (num != 0);
```

Lệnh break

- Lệnh break có thể xuất hiện bên trong vòng lặp (while, do, hay for) hoặc một lệnh switch.
- Nó gây ra bước nhảy ra bên ngoài những lệnh này và vì thế kết thúc chúng.
- Giống như lệnh continue, lệnh break chỉ áp dụng cho vòng lặp hoặc lệnh switch gần nó nhất.
- Không sử dụng lệnh break bên ngoài vòng lặp hay lệnh switch.

Ví dụ: Viết chương trình tính tổng các số nguyên được nhập từ bàn phím, chương trình được kết thúc khi nhập số âm.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int  so,tong=0;
    clrscr();
    while (1)
    {
          cout <<"Nhap mot so nguyen: ";
          cin >>so;
          if (so<0)  break;
          tong +=so;
    }
    cout <<"\nTong cac gia tri da nhap la: "<<tong;
    getch();
    return;
}
```

Chương trình trên nếu không dùng break có thể được viết lại như sau:

```
do
{
        cout <<"Nhap mot so nguyen: ";
        cin >>so;
        if (so>=0)
                tong +=so;
}while (so>=0);
```

```
#include<iostream.h>
#include<math.h>
void main()
{
   int n; char chon;
   cout<<“n= ”; cin>>n;
   cout<<“Luy thua tu 0 den 10 cua n: \n”;
   for (int i=0;i<=10;i++)
   {
        cout<<n<<“^”<<i<<“= ”;
        cout<<pow(n,i);
        cout<<“Nhap Q de thoat, nhap 1 phim bat ky de tiep
   tuc.”;
        cin>>chon;
        if (chon==‘Q’) break;
   }
}
```

Lệnh goto

- Lệnh goto cung cấp một hình thức nhảy tự do không có cấu trúc

Goto *Nhãn;*

Lệnh goto

```
for (i = 0; i < attempts; ++i)
{
    cout << "Please enter your password: ";
    cin >> password;
    if (Verify(password)) // check password for correctness
    goto out; // drop out of the loop
    cout << "Incorrect!\n";
}
out:
//etc...
```

Lệnh return

Lệnh return cho phép một hàm trả về một giá trị cho thành phần gọi nó.

Cú pháp:

return *biểu thức*;

- Trong đó *biểu thức* chỉ rõ giá trị được trả về bởi hàm.
- Kiểu của giá trị này phải hợp với kiểu của hàm.
- Trường hợp kiểu trả về của hàm là **void,** *biểu thức* sau lệnh *return* rỗng:

```
return;
```

Lệnh return

```
int main (void)
{
    cout << "Hello World\n";
    return 0;
}
```

int