

# Bài 2. Cấu trúc điều khiển

*Giảng viên:*

# Hệ thống bài cũ

- Một số khái niệm
- Môi trường làm việc
- Các bước phát triển phần mềm
- Cấu trúc một chương trình C++
- Bảng ký tự của C++
- Các từ khóa
- Tên gọi trong C++
- Một số quy tắc viết mã lệnh
- Lệnh xuất-nhập
- Kiểu dữ liệu cơ bản, biến, hằng

# MỤC TIÊU

- Hiểu và sử dụng các cấu trúc điều khiển

# Nội dung

- Giới thiệu các cấu trúc điều khiển
- Giới thiệu chi tiết lệnh if và if-else
- Biểu thức điều kiện
- Giới thiệu chi tiết lệnh switch
- Giới thiệu chi tiết lệnh do...while
- Giới thiệu chi tiết lệnh while
- Giới thiệu chi tiết lệnh for

# Các Cấu Trúc Điều Khiển

Tất cả chương trình chỉ có thể được viết dưới ba dạng cấu trúc điều khiển:

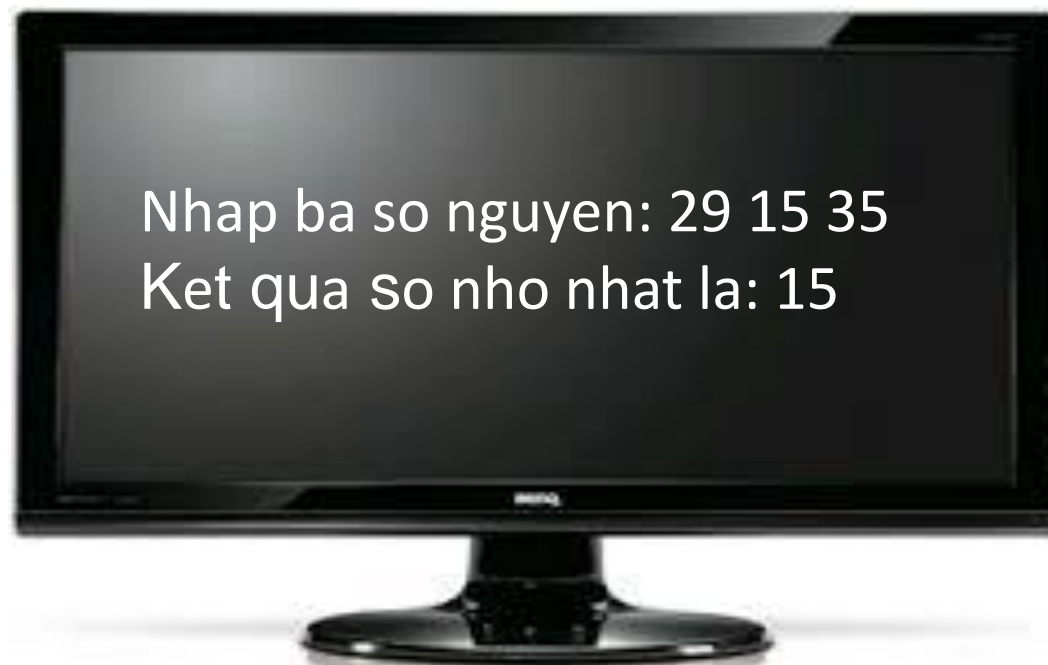
- Cấu trúc tuần tự (mặc định).
- Cấu trúc lựa chọn (có 3 loại): **if**, **if-else**, **switch** .
- Cấu trúc lặp (có 3 loại): **for**, **while**, **do-while** .

# Lệnh if

## Vấn đề

Tìm số nhỏ nhất trong ba số nguyên được nhập từ bàn phím.

## Thực hiện



# Giải Thuật

- Đưa thông báo nhắc người sử dụng nhập vào ba số nguyên.
- Đọc các giá trị nguyên và lần lượt gán vào các biến  $x$ ,  $y$ , và  $z$ .
- Khai báo biến **nhonhat** và gán **nhonhat** =  $x$ .
- Nếu giá trị của **nhonhat** lớn hơn  $y$  thì gán **nhonhat** =  $y$ .
- Nếu giá trị của **nhonhat** lớn hơn  $z$  thì gán **nhonhat** =  $z$ .
- Đưa giá trị của **nhonhat** ra màn hình.

# Viết mã, thực thi và kiểm tra

- Tạo khung chương trình
  - Tạo tập tin chương trình nguồn.
  - Thêm vào các chỉ dẫn biên dịch.
  - Thêm vào hàm main().
- Chuyển đổi từng bước giải thuật thành mã lệnh
  - Thêm các khai báo cho những đối tượng chưa được khai báo.
  - Khai báo bao gồm tên và kiểu dữ liệu.
- Xem mã nguồn và kết quả



GiaTriNhoNhat3SoNguyen.cpp

(Global Scope)

main()

```
# include <iostream>
using namespace std;
//chuong trinh in gia tri nho nhat 3 so nguyen khi nhap vao
int main()
{
    //nhap ba so nguyen
    int m,n,k;
    cout << "Moi ban nhap 3 so nguyen m,n,k: " << endl;
    cin >> m >> n >> k;
    int min = m;    // lúc này min <= n1
    if (n < min) min = n;    // lúc này min <= n1, n2
    if (k < min) min = k;    // lúc này min <= n1, n2, n3

    cout << "Gia tri nho nhat trong 3 so la: " << min << endl;

    system ("pause");
    return 0;
}
```

C:\Users\hieunh3\Document...

```
Moi ban nhap 3 so nguyen m,n,k:
29
15
35
Gia tri nho nhat trong 3 so la: 15
Press any key to continue . . .
```

## Lệnh if

Lệnh **if** cho phép thực thi có điều kiện.

cú pháp : **if** (*condition*) *statement*;

*condition* là một biểu thức nguyên và *statement* là một lệnh thực thi bất kỳ. Lệnh này sẽ được thực thi chỉ khi giá trị của biểu thức nguyên đó là khác không. Lưu ý các dấu đóng mở ngoặc là bắt buộc bao quanh biểu thức điều kiện này.

**Ví dụ:**

```
int main( )
{
    int n, d;
    cout << "Enter two positive integers: ";
    cin >> n >> d;
    if (n%d) cout << n << " is not divisible by " << d << endl;
}
```

Khai báo nào dưới đây cho thấy cú pháp đúng của câu lệnh if ?

if condition

if { condition

if (condition) statement;

condition if



DEMO

Lệnh if



# Lệnh if..else

## Vấn đề

Viết chương trình giải phương trình bậc nhất  $ax + b = 0$

## Thực hiện



## Giải Thuật

- Đưa thông báo nhắc người sử dụng nhập vào hệ số  $a, b$ .
- Đọc các giá trị thực và lần lượt gán vào các biến  $a, b$ .
- Nếu  $a = 0$  thì xét tiếp:
  - Nếu  $b = 0$ , xuất "PT có vô số nghiệm".
  - Ngược lại,  $b \neq 0$ , xuất "PT vô nghiệm".
- Ngược lại, nếu  $a \neq 0$  thì hiển thị nghiệm là:  $x = -b/a$

# Viết mã, thực thi và kiểm tra

- Tạo khung chương trình
  - Tạo tập tin chương trình nguồn.
  - Thêm vào các chỉ dẫn biên dịch.
  - Thêm vào hàm main( ).
- Chuyển đổi từng bước giải thuật thành mã lệnh
  - Thêm các khai báo cho những đối tượng chưa được khai báo.
  - Khai báo bao gồm tên và kiểu dữ liệu.
- Xem mã nguồn và kết quả

PhuongTrinhBac1.cpp

(Global Scope)

main()

```
# include <iostream>
using namespace std;
//giai phuong trinh bac 1
int main()
{
    //nhap hai so phuong trinh
    double a,b;
    cout << "Moi ban nhap he so phuong trinh a,b: " << endl;
    cin >> a >> b;
    if (a==0){
        if (b!=0){
            cout << "PT vo nghiem";
        }
        else {
            cout <<"PT co vo so nghiem";
        }
    }
    else{
        cout << "Nghiem phuong trinh = "<< -b/a;
    }
    cout <<endl;
    system ("pause");
    return 0;
}
```

C:\Users\hieunh3\Documents\...

```
Moi ban nhap he so phuong trinh a,b:
3
9
Nghiem phuong trinh = -3
Press any key to continue . . .
```



## Lệnh if..else

Lệnh **if..else** đưa ra hai lệnh tùy chọn để thực hiện tùy thuộc vào giá trị biểu thức điều kiện là đúng hay sai.

Cú pháp :

```
if (condition) statement1;  
else statement2;
```

Với *condition* là một biểu thức nguyên, *statement1*, *statement2* là các lệnh thực thi. Nếu giá trị của biểu thức điều kiện là khác không thì *statement1* sẽ được thực hiện; trái lại *statement2* sẽ được thực hiện.

# Các Lệnh Lựa Chọn Lồng Nhau

Là lệnh `if-else` nhưng phần `if` hay `else` của nó là câu lệnh `if-else` khác:

```
if(boolean_exp1)  
    statement1
```

```
else  
    if(boolean_exp2)  
        statement2  
    else if ...
```

# Các Lệnh Lựa Chọn Lồng Nhau

Cấu trúc **else if**

Các lệnh **if..else** thường được dùng để kiểm tra một chuỗi lựa chọn song song, trong đó chỉ các mệnh đề **else** chứa các lệnh lồng tiếp theo.

Trong trường hợp này, lệnh kết hợp thường được định dạng bằng cách xếp các cụm **else if** thẳng hàng để nhấn mạnh tính song song của cấu trúc logic.

# Các Lệnh Lựa Chọn Lồng Nhau - else if

```
int main( )
{
    char language;
    cout << "Engl., Fren., Ger., Ital., or Rus.? (e|f|g|i|r): ";
    cin >> language;
    if (language == 'e') cout << "Welcome to ProjectEuclid.";
    else if (language == 'f') cout << "Bon jour, ProjectEuclid.";
    else if (language == 'g') cout << "Guten tag, ProjectEuclid.";
    else if (language == 'i') cout << "Bon giorno, ProjectEuclid.";
    else if (language == 'r') cout << "Dobre utre, ProjectEuclid.";
    else cout << "Sorry; we don't speak your language.";
}
```

Cho biết kết quả của chương trình sau?

```
#include <iostream>
using namespace std;
void main( )
{
    int a=3,b=5;
    if (a<b)
        a=a+1;
    else
        b=b+1;
    a=a+b;
    cout<<a;
}
```

4

14

9

3



**DEMO**

Lệnh else ... if



# Toán tử biểu thức điều kiện

- C++ cung cấp một toán tử đặc biệt thường được dùng thay cho lệnh `if...else`, được gọi là **toán tử biểu thức điều kiện**.
- Nó sử dụng ký tự dấu hỏi (?) và dấu hai chấm (:) trong **cú pháp**:  
*condition ? expression1 : expression2*

- **Ví dụ:**

```
int main()  
{  
    int m, n;  
    cout << "vi du mo ta toan tu bieu thuc dieu kien:" << endl;  
    cout << "Moi ban nhap 2 so m, n: ";  
    cin >> m >> n;  
    cout << ( m<n ? m : n ) << " la so nho nhat." << endl;  
}
```

# Lệnh rẽ nhánh switch

## Vấn đề

Giả sử sinh viên thi lái xe oto (thang điểm 10) dựa vào ba loại điểm:

Điểm đi học = 10%.

Điểm thi lý thuyết = 40%.

Điểm thi thực hành = 50%.

Chương trình sẽ đọc các điểm này và tính hạng của sinh viên.

Biết rằng, nếu:

Điểm  $\geq 9$ : Hạng A.

$8 \leq \text{điểm} < 9$ : Hạng B.

$7 \leq \text{điểm} < 8$ : Hạng C.

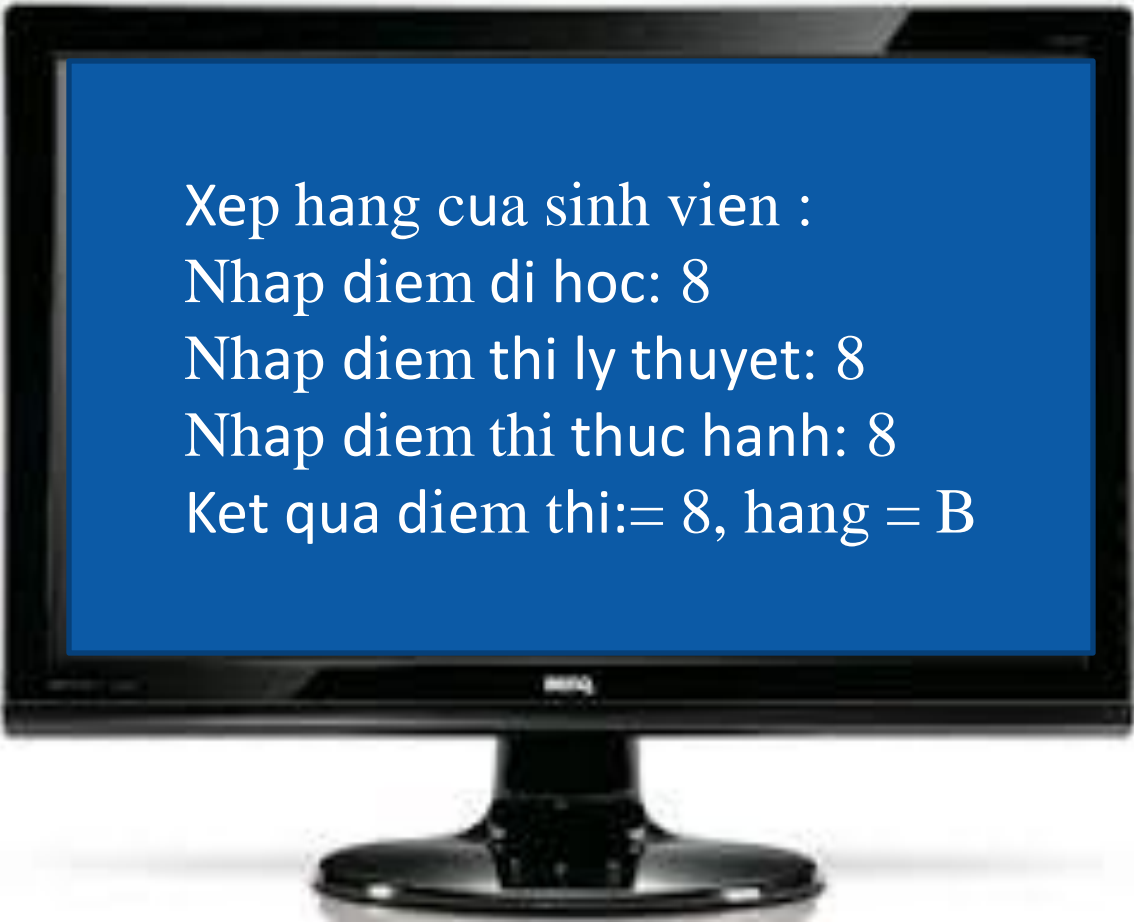
$6 \leq \text{điểm} < 7$ : Hạng D.

Điểm  $< 6$ : Hạng F.



# Lệnh rẽ nhánh switch

## Thực hiện



Xếp hạng của sinh viên :  
Nhập điểm di học: 8  
Nhập điểm thi lý thuyết: 8  
Nhập điểm thi thực hành: 8  
Kết quả điểm thi:= 8, hạng = B

# Giải Thuật

- Khai báo các hằng: `tyle_d_dihoc`, `tyle_d_lythuyet`, `tyle_d_thuchanh` ứng với tỉ lệ đã cho: 10%, 40%, 50%.
- Nhắc người dùng nhập điểm đi học, lý thuyết, thực hành
- Nhập điểm là các số thực tương ứng với các biến: `d_dihoc`, `d_lythuyet`, `d_thuchanh`
- Tính kết quả điểm cuối cùng:  $KQ = \text{tyle\_d\_dihoc} * d\_dihoc + \text{tyle\_d\_lythuyet} * d\_lythuyet + \text{tyle\_d\_thuchanh} * d\_thuchanh$
- Hiển thị hạng tương ứng với KQ điểm cuối kỳ

# Viết mã, thực thi và kiểm tra

- Tạo khung chương trình
  - Tạo tập tin chương trình nguồn.
  - Thêm vào các chỉ dẫn biên dịch.
  - Thêm vào hàm main( ).
- Chuyển đổi từng bước giải thuật thành mã lệnh
  - Thêm các khai báo cho những đối tượng chưa được khai báo.
  - Khai báo bao gồm tên và kiểu dữ liệu.
- Xem mã nguồn và kết quả

DiemThiLaiXe.cpp

(Global Scope)

main()

```
# include <iostream>
using namespace std;
//Xep hang Diem cho viec thi lai xe oto
int main()
{
    //khai bao ty le diem thi
    const double tl_d_dihoc=0.1;
    const double tl_d_lythuyet=0.4;
    const double tl_d_thuchanh=0.5;

    //khai bao bien diem thi
    double d_dihoc,d_lythuyet,d_thuchanh;
    //khai bao bien KQ
    int KQ;
    cout << "Moi ban nhap diem di hoc: "; cin >> d_dihoc;cout <<endl;
    cout << "Moi ban nhap diem ly thuyet: ";cin >> d_lythuyet; cout <<endl;
    cout << "Moi ban nhap diem thuc hanh: ";cin >> d_thuchanh; cout <<endl;
    KQ =(int)( tl_d_dihoc*d_dihoc + tl_d_lythuyet*d_lythuyet + tl_d_thuchanh*d_thuchanh);
    switch ((int)KQ)
    {
        case 10: cout <<"Diem ban: " << KQ << " tuong ung hang A" << endl; break;
        case 9: cout <<"Diem ban: " << KQ << " tuong ung hang A" << endl; break;
        case 8: cout <<"Diem ban: " << KQ << " tuong ung hang B" << endl; break;
        case 7: cout <<"Diem ban: " << KQ << " tuong ung hang C" << endl; break;
        case 6: cout <<"Diem ban: " << KQ << " tuong ung hang D" << endl; break;
        case 5: case 4: case 3: case 2: case 1: case 0: cout <<"Diem ban: " << KQ << " tuong ung hang F" << endl; break;
        default: cout << "Error: Diem ngoai khoang.\n";
    }
    system ("pause");
    return 0;
}
```

C:\Users\hieunh3\Documen...

```
Moi ban nhap diem di hoc: 8
Moi ban nhap diem ly thuyet: 8
Moi ban nhap diem thuc hanh: 8

Diem ban: 8 tuong ung hang B
Press any key to continue . . .
```

# Lệnh rẽ nhánh switch

Lệnh **switch** có thể được sử dụng thay cấu trúc **else if** được sử dụng khi có nhiều nhánh rẽ phụ thuộc vào giá trị của cùng một expression

## Cú pháp:

switch (*expression*)

```
{  
  case constant1 : statementList1;  
  case constant2 : statementList2;  
  case constant3 : statementList3;  
  :  
  case constantN : statementListN;  
  default : statementList0;  
}
```

Biểu thức **int** hay **char**

Mỗi *statementList* được kết thúc bằng lệnh **break** hay **return**

Mỗi **hằng số** constant cũng là **int** hay **char**

```
#include <iostream>
using namespace std;
main()
{
    switch (5) {
        case 5: cout<<" 5 ";
        default: cout<<" 10 ";
        case 6: cout<<" 6 ";
    }
}
```

Kết quả:

5 10 6

5 6


5

10



# DEMO

Câu lệnh SWITCH

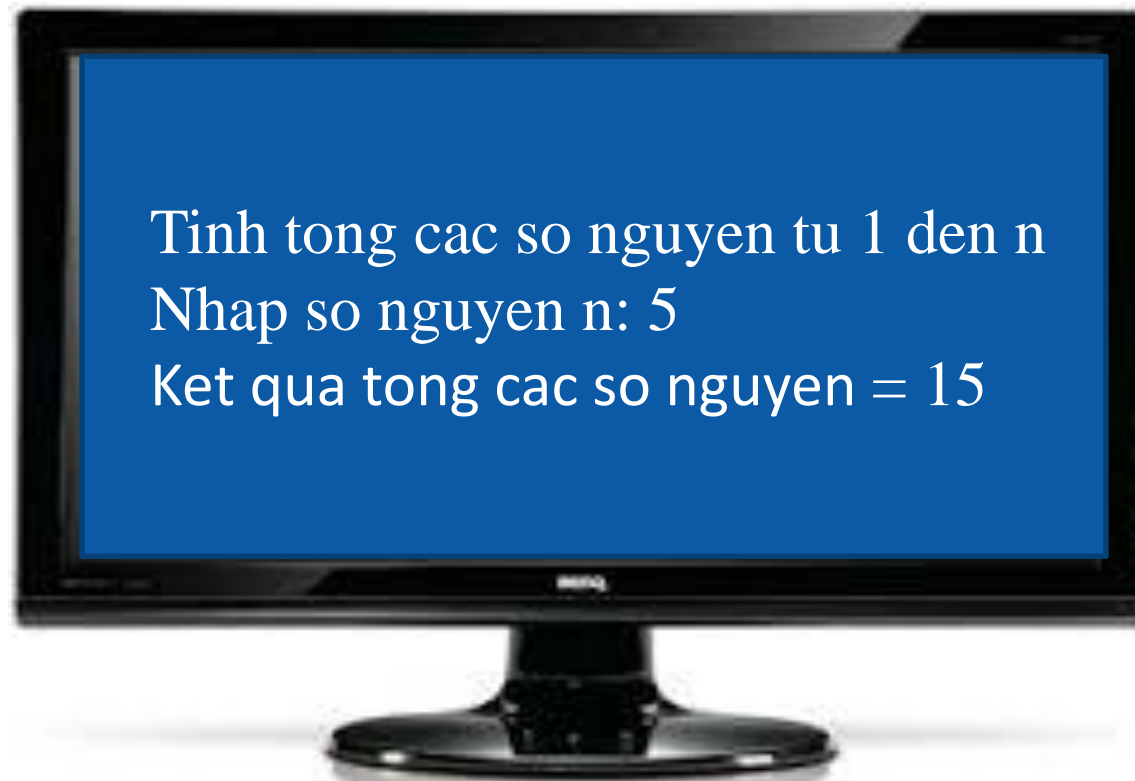
A white hand cursor icon, resembling a computer mouse pointer, is pointing at the letter 'O' in the word 'DEMO'. The hand is stylized with a white outline and a white fill.

# Lệnh lặp- While

## Vấn đề

Muốn tính tổng của tất cả các số nguyên từ 1 đến n.

## Thực hiện





## Giải Thuật

- Xuất thông báo nhắc người sử dụng nhập một số nguyên bất kỳ.
- Đọc giá trị thực từ bàn phím vào biến **songuyen** .
- Tính tổng các số nguyên từ 1 đến songuyen:  
$$\text{tong} = 1 + 2 + \dots + \text{songuyen};$$
- Xuất kết quả tổng các số nguyên: **tong** .

# Viết mã, thực thi và kiểm tra

- Tạo khung chương trình
  - Tạo tập tin chương trình nguồn.
  - Thêm vào các chỉ dẫn biên dịch.
  - Thêm vào hàm main( ).
- Chuyển đổi từng bước giải thuật thành mã lệnh
  - Thêm các khai báo cho những đối tượng chưa được khai báo.
  - Khai báo bao gồm tên và kiểu dữ liệu.
- Xem mã nguồn và kết quả

```
# include <iostream>
using namespace std;
//Tinh tong cac so nguyen tu 1 den n
int main()
{
    //khai bao bien so nguyen nhap vao va bien tong
    int songuyen,Tong=0;
    cout <<"Chuong trinh tinh tong cac so nguyen tu 1 den n !" <<endl;
    cout << "Moi ban nhap so nguyen n: ";    cin >>  songuyen;cout <<endl;
    int i=1;
    while (i<=songuyen)
    {
        Tong +=i;
        i++;
    }
    cout <<"Tong cua tat ca so nguyen tu 1 den  " <<songuyen <<" bang: " <<Tong << endl;
    system ("pause");
    return 0;
}
```

C:\Users\hieunh3\Documents\Visual Studi...

```
Chuong trinh tinh tong cac so nguyen tu 1 den n !
Moi ban nhap so nguyen n: 5

Tong cua tat ca so nguyen tu 1 den  5 bang: 15
Press any key to continue . . . _
```

# Lệnh lặp- While

**Cú pháp :** **while** (*condition*) *statement*;

*condition* là một biểu thức kiểu nguyên.

*statement* là một lệnh thực thi bất kỳ.

Nếu giá trị của **biểu thức** bằng 0 (nghĩa là "sai") thì *statement* sẽ bị bỏ qua và chương trình sẽ thực hiện lệnh tiếp theo sau khối lệnh **while**.

Nếu giá trị của **biểu thức** khác 0 (nghĩa là "đúng") thì *statement* sẽ được thực thi lặp lại cho tới khi nào biểu thức có giá trị bằng 0.

Lưu ý là *condition* phải được đóng trong các dấu ngoặc đơn.

# Kết Thúc Vòng Lặp

- Chúng ta đã thấy cách mà lệnh **break** được dùng để điều khiển lệnh **switch**.
- Ngoài ra, lệnh **break** cũng được dùng để điều khiển các vòng lặp.
- Phím tắt thoát khỏi vòng lặp vô hạn: **ctrl** + **c**.

## Ví dụ:

```
int main ( )
{
    int n, i=1;
    cout << "Nhap so nguyen n: ";
    cin >> n; long sum=0;
    while (true)
    {
        if (i > n) break;    // kết thúc vòng lặp ngay lập tức
        sum += i++;
    }
    cout << "Tong cua so nguyen: " << n << " la: " << sum;
}
```

Phím tắt nào cho phép thoát khỏi vòng lặp vô hạn?

Ctrl + e

Ctrl + c

Ctrl + Shift

Alt + c



**DEMO**

**Lệnh WHILE**



## LỆNH do..while

Theo ví dụ cho lệnh **while** về tính tổng của tất cả các số nguyên từ 1 đến n.

```
int main( )
{
    int n, i=0;
    cout << "Nhap so nguyen n: ";
    cin >> n; long sum=0;
    do sum += i++;
    while (i <= n);
    cout << "Tong cua so nguyen " << n << " la: " << sum;
}
```



# LỆNH **do..while**

**Cú pháp:** **do** *statement* **while** (*condition*);

*condition* là một biểu thức nguyên

*statement* là một lệnh thực thi bất kỳ.

- Nó lặp việc thực thi lệnh *statement* và sau đó kiểm tra giá trị biểu thức *condition* cho tới khi nào giá trị biểu thức là false.
- Lệnh **do..while** cũng làm việc tương tự lệnh **while** ngoại trừ việc biểu thức điều kiện được kiểm tra ở cuối vòng lặp thay vì ở đầu vòng lặp.
- Điều này có nghĩa là bất kỳ biến kiểm soát vòng lặp nào cũng có thể được định nghĩa trong vòng lặp thay vì trước nó.
- Ngoài ra, một vòng lặp **do...while** sẽ lặp ít nhất một lần bất kể giá trị của biểu thức điều kiện là đúng hay sai.

Số lần nhỏ nhất của một vòng lặp do..while ?

-1

0

1

2



**DEMO**

**DEMO Lệnh do..while**



# LỆNH for

Theo ví dụ cho lệnh **while** về tính tổng của tất cả các số nguyên từ 1 đến n.

```
int main()
{
    int n;
    cout << "Nhap mot so nguyen n: "; cin >> n;
    long sum=0;
    for (int i=1; i < n/2; i++) // phạm vi của biến i là ở trong vòng lặp này
        sum += i;
    for (int i=n/2; i <= n; i++) // phạm vi của biến i là ở trong vòng lặp này
        sum += i;
    cout << "Tong so nguyen " << n << " la: " << sum << endl;
}
```

# LỆNH for

## Cú pháp lệnh for:

**for** (*initialization*; *condition*; *update*) *statement*;

*initialization*, *condition*, và *update* là các biểu thức tùy chọn, và *statement* là lệnh thực thi bất kỳ.

Bộ ba (*initialization*; *condition*; *update*) sẽ điều khiển vòng lặp.

Trình tự các bước của một vòng lặp là:

1. thực hiện biểu thức *initialization*;
2. nếu giá trị của biểu thức *condition* là sai thì kết thúc lặp;
3. thực hiện lệnh *statement*;
4. thực hiện biểu thức cập nhật *update*;
5. lặp lại các bước từ 2 đến 4.



**DEMO**

**DEMO Lệnh for**

A white hand cursor icon, resembling a computer mouse pointer, is positioned to the right of the text. The index finger is pointing upwards towards the 'O' in 'DEMO'.

# LỆNH break

Chúng ta đã thấy cách sử dụng lệnh **break** bên trong lệnh **switch**.

Nó cũng được sử dụng bên trong vòng lặp. Khi lệnh này được thực hiện, nó sẽ kết thúc vòng lặp tức là “**thoát ra ngoài**” vòng lặp ngay lúc đó.

**Ví dụ:**

```
int main( )
{
    int n, i=1;
    cout << "Enter a positive integer: "; cin >> n;
    long sum=0;
    while (true)
    {
        if (i > n) break;
        sum += i++;
    }
    cout << "The sum of the first " << n << " integers is " << sum;
}
```

Enter a positive integer: **8**

The sum of the first 8 integers is 36

# LỆNH **continue**

Lệnh **break** giúp hủy phần lệnh còn lại trong khối lặp và nhảy ngay lập tức sang lệnh tiếp theo bên ngoài vòng lặp.

Lệnh **continue** cũng tương tự như vậy.

## **Ví dụ:**

```
int main()
{
    int n;
    for (;;)
    {
        cout << "Enter int: "; cin >> n;
        if (n%2 == 0) continue;
        if (n%3 == 0) break;
        cout << "\tBottom of loop.\n";
    }
    cout << "\tOutside of loop.\n";
}
```



# LỆNH goto

- Lệnh **break**, **continue**, và lệnh **switch** đều cho phép trình điều khiển chương trình rẽ nhánh đến các vị trí khác với vị trí được lập trình trước theo một lộ trình thông thường. Đích của lệnh rẽ nhánh được quyết định bởi ngữ cảnh: lệnh **break** đi tới lệnh tiếp theo bên ngoài vòng lặp, **continue** đi tới lệnh kiểm tra điều kiện tiếp theo, và **switch** đi tới trường hợp đúng với các hằng số đã được xác định trước.
- Cả ba lệnh trên được gọi là *lệnh rẽ nhánh* bởi vì chúng khiến cho trình điều khiển chương trình “nhảy qua” các lệnh khác.
- Lệnh **goto** là một loại khác của lệnh rẽ nhánh. Đích đến của nó được cụ thể bằng một **nhãn** gán trong lệnh.
- Một *nhãn* đơn giản chỉ là một định danh theo sau bởi một dấu hai chấm đặt trước một lệnh.
- Các **nhãn** làm việc giống như các lệnh **case** trong lệnh **switch** chúng đưa ra địa chỉ của lệnh rẽ nhánh.

# LỆNH goto

```
int main()
{
    const int N=5;
    for (int i=0; i<N; i++)
    {
        for (int j=0; j<N; j++)
        {
            for (int k=0; k<N; k++)
            if (i+j+k>N) goto lblesc;
            else cout << i+j+k << " ";
            cout << "* ";
        }
        lblesc : cout << "." << endl; //trong vòng lặp i, ngoài vòng lặp j
    }
}
```



**DEMO**

Các lệnh  
`break; goto; continue`

A white hand cursor icon, resembling a computer mouse pointer, is positioned below the word 'DEMO'. The index finger is pointing upwards towards the 'O' in 'DEMO'.

# Sinh các số giả ngẫu nhiên

## Ví dụ

```
#include <cstdlib> // định nghĩa hàm rand() và hằng số RAND_MAX
#include <iostream>
using namespace std;
int main( )
{ // in ra các số giả ngẫu nhiên:
  for (int i = 0; i < 8; i++)
    cout << rand() << endl;
    cout << "RAND_MAX = " << RAND_MAX << endl;
}
```

Chương trình này sử dụng hàm `rand()` để sinh các số giả ngẫu nhiên.

# Tổng kết

- Giới thiệu các cấu trúc điều khiển
- Giới thiệu chi tiết lệnh if và if-else
- Biểu thức điều kiện
- Giới thiệu chi tiết lệnh switch
- Giới thiệu chi tiết lệnh do...while
- Giới thiệu chi tiết lệnh while
- Giới thiệu chi tiết lệnh for

## Chuẩn bị bài sau

Sinh viên đọc sách và slide trước bài học kế tiếp về Hàm gồm:

- Giới thiệu
- Định nghĩa hàm, gọi hàm.
- Nguyên mẫu hàm.
- Các bước thiết kế hàm.
- Phạm vi.
- Hàm không trả trị.
- Một số hàm toán học.
- Xét chi tiết cách truyền tham số.
- Hàm nội tuyến - hàm inline.
- Nạp chồng hàm.
- Hàm `main()`, hàm nhị phân



**FPT POLYTECHNIC**

THANK YOU!

[www.poly.edu.vn](http://www.poly.edu.vn)