



NGÔN NGỮ LẬP TRÌNH C++

§2

- Câu lệnh rẽ nhánh: `if ()`
- Câu lệnh rẽ nhánh: `if() - else`
- Khối lệnh: `{ }`
- Các phép toán so sánh:

A. Câu lệnh rẽ nhánh `if ()`

1. Rẽ nhánh trong xử lý

Ta nhận thấy, có nhiều trường hợp, một việc nào đó chỉ xảy ra khi có một điều kiện tương ứng xảy ra:

Ví dụ:

Hôm nay Trời mưa nên Đức đi học phải mang áo mưa. Như vậy, sự việc “Đức đi học phải mang áo mưa” chỉ xảy ra khi “Trời mưa”.

2. Câu lệnh `if ()`

Cú pháp:

```
if (<Điều kiện>
    <Câu lệnh>
```

Thực hiện:

Nếu <Điều kiện> đúng thì thực hiện <Câu lệnh>.

Ví dụ 1: Chương trình nhập vào hai số nguyên a, b và so sánh hai số nguyên đó.

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int a, b;
4  int main() {
5      cout<<"Nhập số a: ";
6      cin>>a;
7      cout<<"Nhập số b: ";
8      cin>>b;
9      if( a < b )
10         cout<<"a là số nhỏ hơn ";
11      if( b < a )
12         cout<<"b là số nhỏ hơn ";
13      if ( a == b )
14         cout<<"a bằng b ";
15 }
```

Ví dụ 2: Chương trình nhập vào một số nguyên a và đưa ra thông báo số nhập vào là số chẵn hay số lẻ.

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int a;
4  int main()
5  {
6      cin>>a;
7      if(a % 2 == 0)
8         cout<<"a là số chẵn";
9      if ( a % 2 != 0 )
10         cout<<"a là số lẻ";
11 }
12
```



B. Câu lệnh rẽ nhánh if() - else

☞ Cú pháp:

```
if (<Điều kiện>
    <Câu lệnh 1> ;
else
    <Câu lệnh 2>;
```

☞ Thực hiện:

Nếu <Điều kiện> đúng thì thực hiện <Câu lệnh 1>; ngược lại, nếu điều kiện sai thì thực hiện <Câu lệnh 2>;


Ví dụ 3: Chương trình nhập vào một số nguyên a và đưa ra thông báo số nhập vào là số chẵn hay số lẻ (sử dụng câu lệnh **if() - else**)

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int a;
4  int main()
5  {
6      cin>>a;
7      if(a % 2 == 0)
8          cout<<"a la so chan";
9      else
10         cout<<"a la so le";
11 }
12
```

C. Khối lệnh { }

Khối lệnh được tạo bởi bằng nhiều câu lệnh nhỏ hơn và được bỏ vào cặp ngoặc { }.

```
{
    <Câu lệnh 1>;
    <Câu lệnh 2>;
    .....
    <Câu lệnh n>;
}
```



Chú ý:

Khi thực hiện khối lệnh, các lệnh bên trong đều được thực hiện.

Ví dụ 4: Hãy chạy chương trình với nhập:

☞ Lần chạy 1: Nhập $a = 10$;

☞ Lần chạy 2: Nhập $a = 11$.



```
1 #include<bits/stdc++.h>
2
3 using namespace std;
4 int a, b, c;
5 int main(){
6     cout<<"Nhap a = ";
7     cin>>a;
8     b = 1;
9     c = 1;
10    if(a % 2 == 0)
11    {
12        b = b + 1;
13        c = c + 1;
14    }
15    cout<<"b = "<<b<<endl;
16    cout<<"c = "<<c<<endl;
17 }
```

Ví dụ 5: Hãy chạy chương trình với nhập:

- ☞ Lần chạy 1: Nhập $a = 10$;
- ☞ Lần chạy 2: Nhập $a = 11$.

```
1 #include<bits/stdc++.h>
2
3 using namespace std;
4 int a, b, c;
5 int main(){
6     cout<<"Nhap a = ";
7     cin>>a;
8     b = 1;
9     c = 1;
10    if(a % 2 == 0)
11    {
12        b = b + 1;
13        c = c + 1;
14    }
15    cout<<"b = "<<b<<endl;
16    cout<<"c = "<<c<<endl;
17 }
```

Hai chương trình khác nhau chỗ nào?

D. Các phép toán so sánh

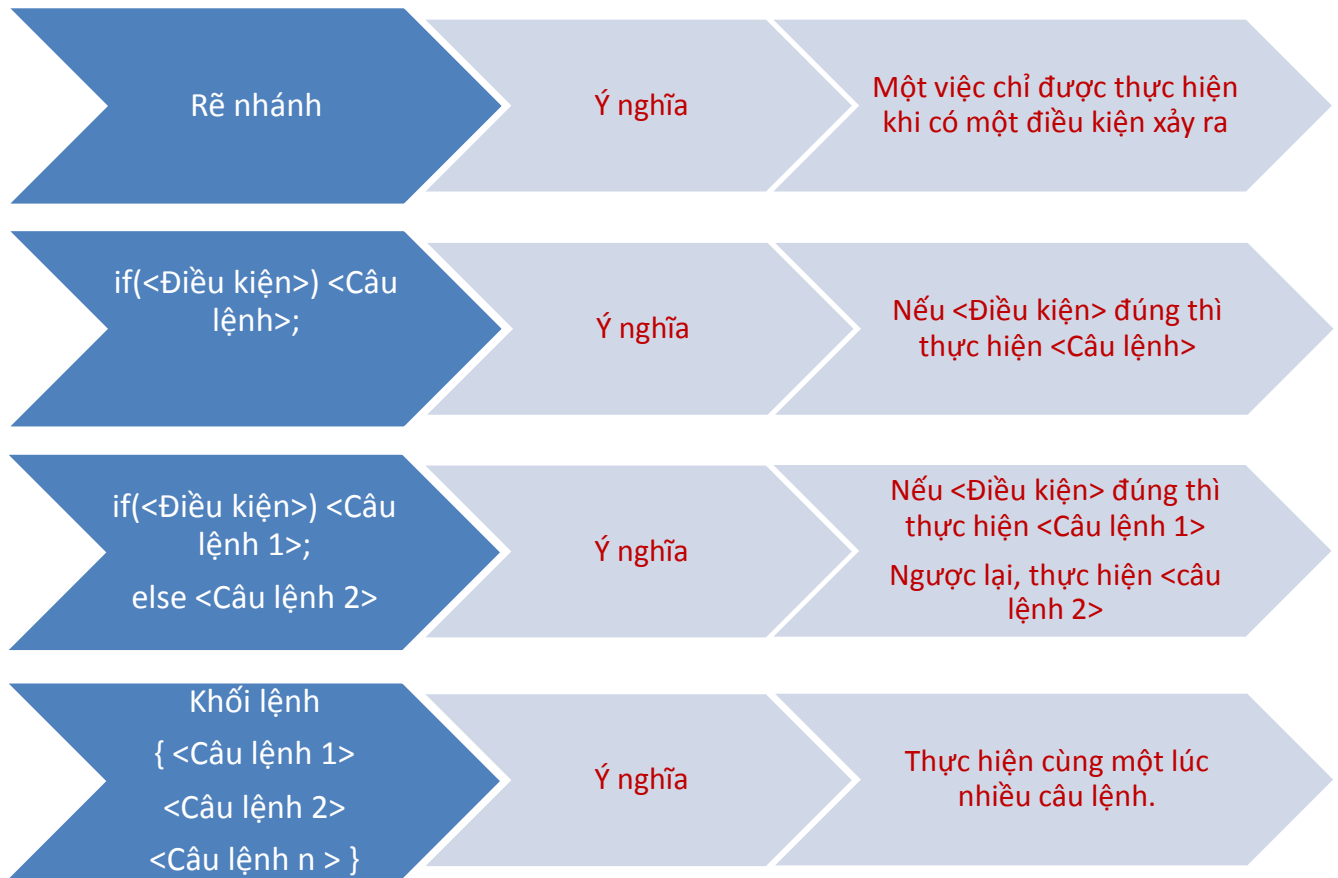
Tên phép toán	Kí hiệu	Ví dụ
So sánh bằng	<code>==</code>	$3 == 3$; ☞ kết quả = true $3 == 4$; ☞ kết quả = false
So sánh khác	<code>!=</code>	$3 != 3$; ☞ kết quả = false
So sánh nhỏ hơn	<code><</code>	$3 < 4$; ☞ kết quả = true
So sánh nhỏ hơn hoặc bằng	<code><=</code>	
So sánh lớn hơn	<code>></code>	
So sánh lớn hơn hoặc bằng	<code>>=</code>	

Chú ý:

Kết quả của phép so sánh là **Đúng** (true) hoặc **Sai** (false).



😊 TÔNG KẾT 😊





BÀI TẬP

**1☀. Kiểm tra chẵn lẻ: ChanLe.CPP**

Viết chương trình nhập vào một số nguyên n . Kiểm tra tính chẵn lẻ của n , tức là nếu n là số lẻ thì đưa ra thông báo “So le”, nếu n là số chẵn thì đưa ra thông báo “So chan”.

Ví dụ:

Input	Output
5	So le
6	So chan

**2☀. Tổng các số lẻ - TongSoLe.Cpp**

Viết chương trình nhập vào 3 số nguyên a, b, c .

Tính tổng các số lẻ được nhập vào.

Ví dụ:

Input	Output
1 2 7	8

**3☀. Biểu thức chẵn – lẻ - BTCL.CPP**

Cho biểu thức $P = a^2 + 2a + b^2 + 3b + 1$.

Viết chương trình nhập vào hai số nguyên a và b . Kiểm tra xem P là số chẵn hay là số lẻ.

Nếu P là số chẵn thì đưa ra thông báo “ P chan”; nếu P là số lẻ thì đưa ra thông báo “ P le”.

Ví dụ:

Input	Output
1 1	P chan
0 1	P le

**4☀. Tổng các số âm - SoAm.CPP**

Viết chương trình nhập vào 5 số nguyên. Tính tổng các số âm (các số nhỏ hơn 0) được nhập vào, đưa ra tổng đó.

Ví dụ:

Input	Output
1 2 -10 3 -20	-30

**5☀. Chữ số lẻ - ChuSoChanTT.CPP**

Viết chương trình nhập số nguyên dương n có hai chữ số: Đưa ra chữ số chẵn của n .

Nếu n có 2 chữ số chẵn, đưa ra 2 chữ số đó, chữ số nhỏ hơn ở dòng 1, chữ số còn lại ở dòng 2.

Nếu n có 1 chữ số chẵn, đưa ra 1 chữ số chẵn đó.

Nếu n không có chữ số lẻ thì đưa ra thông báo: “Khong co chu so chan”.

Ví dụ:



Input	Output
12	2
42	2 4
13	Không có chu số chẵn



6☀. Tổng 2 trong 3 - Sum2in3.Cpp

Cho 3 số nguyên A, B, C và số S . Kiểm tra xem, liệu có thể chọn 2 số trong 3 số A, B, C để tổng của chúng bằng S hay không?

Dữ liệu:

Dòng 1: Nhập 3 số nguyên A, B, C .

Dòng 2: Nhập số nguyên S .

Kết quả:

Ghi 1 nếu chọn được.

Ghi 0 nếu không có cách chọn.

Ví dụ:

Input	Output	Giải thích
1 2 3 5	1	Có thể chọn số 2 và 3 để có tổng bằng 5.
1 2 3 10	0	Không thể chọn 2 số trong 3 số 1, 2, 3 để tổng bằng 10.



7☀. Tổng 2 trong 4 - Sum2in4.Cpp

Cho 4 số nguyên A, B, C, D và số S . Kiểm tra xem, liệu có thể chọn 2 số trong 4 số A, B, C, D để tổng của chúng bằng S hay không?

Dữ liệu:

Dòng 1: Nhập 4 số nguyên A, B, C, D .

Dòng 2: Nhập số nguyên S .

Kết quả:

Ghi 1 nếu chọn được.

Ghi 0 nếu không có cách chọn.

Ví dụ:

Input	Output	Giải thích
1 2 3 4 6	1	Có thể chọn số 2 và 4 để có tổng bằng 6.
1 2 3 4 10	0	Không thể chọn 2 số trong 4 số 1, 2, 3, 4 để tổng bằng 10.

**8☀️ Tổng 3 trong 5 - Sum3in5.Cpp**

Cho 5 số nguyên A, B, C, D, E và số S . Kiểm tra xem, liệu có thể chọn 3 số trong 5 số A, B, C, D, E để tổng của chúng bằng S hay không?

Dữ liệu:

Dòng 1: Nhập 5 số nguyên A, B, C, D, E .

Dòng 2: Nhập số nguyên S .

Kết quả:

Ghi 1 nếu chọn được.

Ghi 0 nếu không có cách chọn.

Ví dụ:

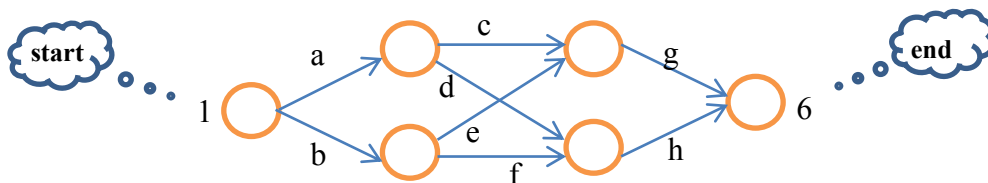
Input	Output	Giải thích
1 2 3 4 7 6	1	Có thể chọn số 1, 2 và 3 để có tổng bằng 6.
1 2 3 4 0 11	0	Không thể chọn 3 số trong 5 số 0, 1, 2, 3, 4 để tổng bằng 11.

**9☀️ Mạng lưới – Net.Cpp**

Cho mạng lưới như hình vẽ. Ta có thể di chuyển từ vị trí 1 đến vị trí 6 bằng cách di chuyển qua các mũi tên, khi đó tổng khoảng cách của việc di chuyển bằng tổng các số ghi trên các mũi tên đó.

Cho biết giá trị a, b, c, d, e, f, g, h .

Yêu cầu: Tìm cách di chuyển từ vị trí 1 đến vị trí 6 với tổng khoảng cách là nhỏ nhất.

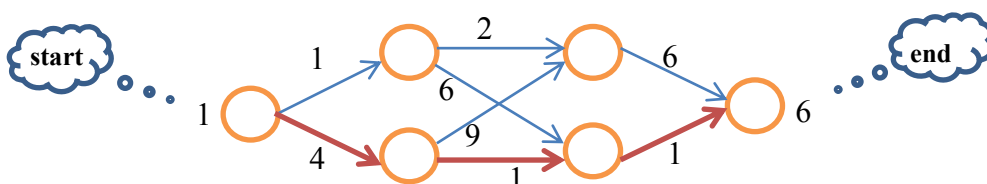


Dữ liệu: Nhập 8 số nguyên dương a, b, c, d, e, f, g, h .

Kết quả: Đưa ra tổng khoảng cách nhỏ nhất khi di chuyển từ vị trí 1 đến vị trí 6.

Ví dụ:

Input	Output
1 4 2 6 9 1 6 1	6

**Giải thích:**

Ta di chuyển từ vị trí 1 đến vị trí 6 bằng các mũi tên tô đậm sẽ có tổng khoảng cách nhỏ nhất.
Bằng $4 + 1 + 1 = 6$.