



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<<"Nhập 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<<"Nhập 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

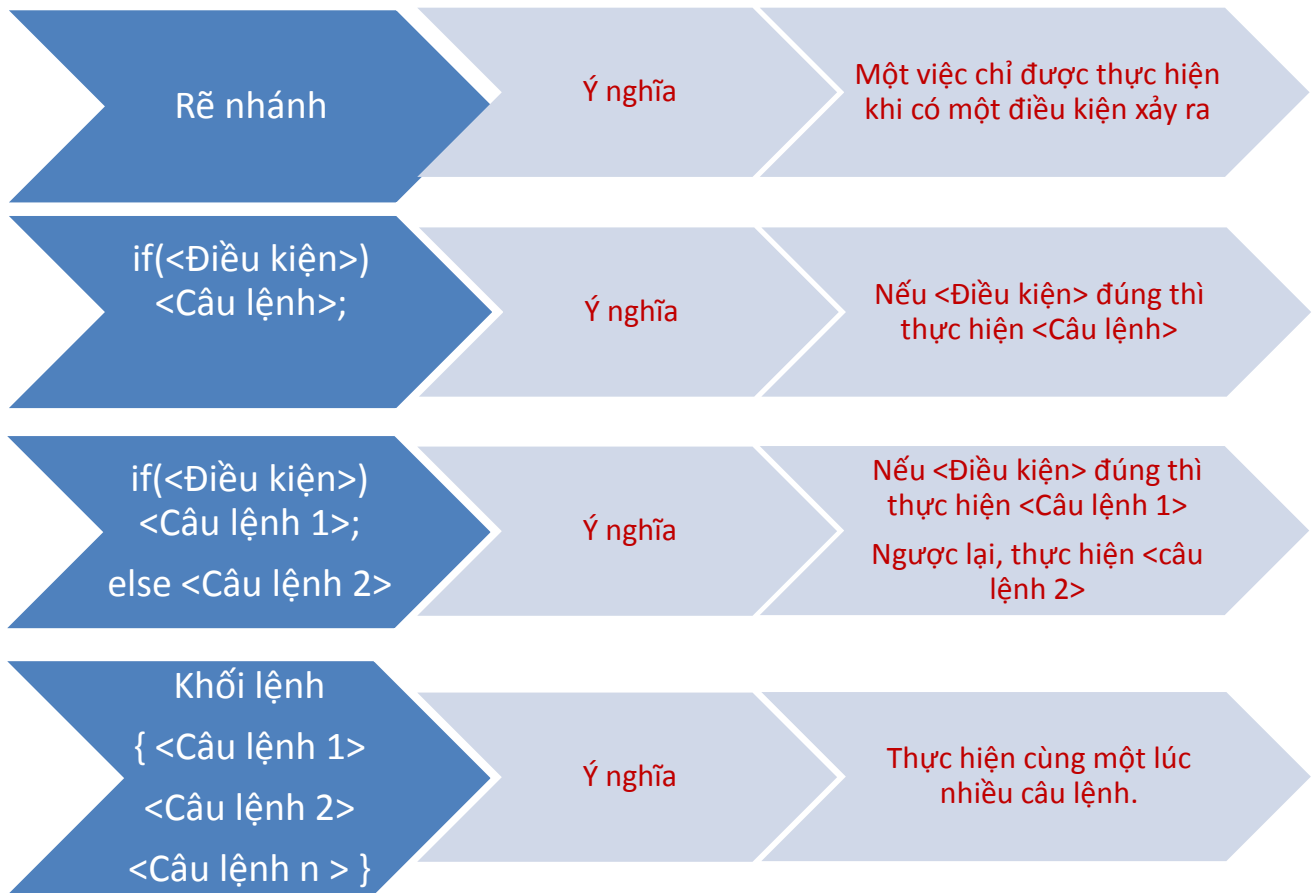
Tên phép toán	Kí hiệu	Ví dụ
So sánh bằng	<code>==</code>	$3 == 3$; ☞ kết quả = true $3 == 4$; ☞ kết quả = false
So sánh khác	<code>!=</code>	$3 != 3$; ☞ kết quả = false
So sánh nhỏ hơn	<code><</code>	$3 < 4$; ☞ kết quả = true
So sánh nhỏ hơn hoặc bằng	<code><=</code>	
So sánh lớn hơn	<code>></code>	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

Viết chương trình nhập vào hai số nguyên khác nhau a và b . Đưa ra số có giá trị lớn hơn.

Ví dụ:

Input	Output
5 6	6
7 2	7



2☀️ Số lớn nhất - Biggist.Cpp

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

Đưa ra số có giá trị lớn nhất.

Ví dụ:

Input	Output
1 2 7	7



3☀️ Số nhỏ nhất - Smallest.Cpp

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

Đưa ra số có giá trị nhỏ nhất.

Ví dụ:

Input	Output
1 2 7	1



4☀️ Tổng các số chia hết cho - DivisorX.Cpp

Viết chương trình nhập vào 3 số nguyên a, b, c và số nguyên dương X . Tính tổng các số trong 3 số a, b, c có giá trị chia hết cho X .

Dữ liệu nhập:

Dòng 1 cho 3 số a, b, c .

Dòng 2 cho số X .

Kết quả là tổng các số chia hết cho X .

Ví dụ:

Input	Output
20 11 30 10	50



5☀️ Số bằng X - EqualX.CPP

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

Kiểm tra xem trong 3 số a, b, c có số nào bằng X hay không?

Dữ liệu nhập:



Dòng 1 cho 3 số a, b, c .

Dòng 2 cho số X .

Kết quả đưa ra 1 nếu có số bằng X , ngược lại, nếu không có số nào bằng X thì đưa ra 0.

Ví dụ:

Input	Output
20 10 30 30	1
20 10 30 150	0

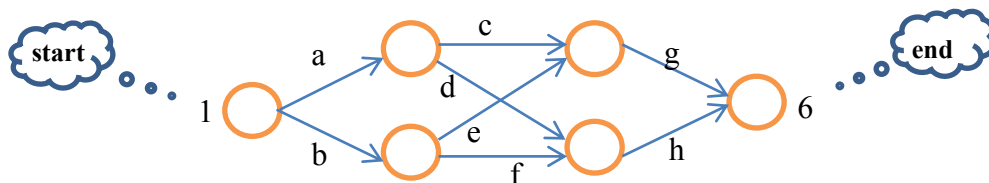


6. Mạng lưới – Net.Cpp

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.

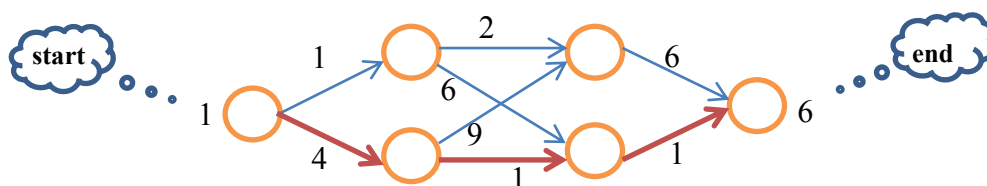


Dữ liệu: Nhập 8 số nguyên dương a, b, c, d, e, f, g, h .

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 $4 + 1 + 1 = 6$.

**7. Đếm số chân - CountLeg.Cpp**

Mẹ Vy đi chợ mua được được N con gà và M con chó. Vy đang học về toán đếm nên liền chạy ra đếm số chân. Hỏi Vy đếm được bao nhiêu cái chân.

Dữ liệu cho trong file **CountLeg.Inp** gồm hai số N và M trên một dòng ($0 < N, M < 10$).

Kết quả ghi ra file **CountLeg.Out** là số chân mà Vy đếm được.

Ví dụ:

CountLeg.Inp	CountLeg.Out
2 1	8

**8. Vừa gà, vừa chó - CockAndDog.Cpp**

Vừa gà vừa chó

Bó lại cho tròn

Ba mươi sáu (36) con

Một trăm (100) chân chẵn

Hỏi mấy gà, mấy chó?

Đây là một bài toán vui nổi tiếng và tìm ra kết quả có nhiều cách. Bây giờ ta sẽ giải bài toán bằng lập trình với bài toán tổng quát hơn:

Tổng số con gà và con chó là X con.

Tổng số chân của các con gà và con chó là Y chân.

Hỏi có bao nhiêu con gà? bao nhiêu con chó?

Dữ liệu cho trong file **CockAndDog.Inp** gồm hai số X và Y .

Kết quả ghi ra file **CockAndDog.Out** gồm hai dòng:

- Dòng đầu ghi số gà tìm được.
- Dòng hai ghi số chó tìm được.

Dữ liệu đảm bảo luôn có nghiệm.

Ví dụ:

CockAndDog.Inp	CockAndDog.Out
36 100	22
	14



**9. Tổng độ lệch hai dãy – BalanceSequence.Cpp**

Cho dãy số A gồm: A_1, A_2, \dots, A_N và dãy B gồm N số hạng B_1, B_2, \dots, B_N . Ta gọi tổng độ lệch hai dãy A và B là tổng $S = |A_1 - B_1| + |A_2 - B_2| + \dots + |A_N - B_N|$.

Yêu cầu: Đưa ra tổng S .

Dữ liệu cho trong file **BalanceSequence.Inp** gồm:

- Dòng đầu ghi số nguyên dương N ($N \leq 100000$);
- Dòng thứ 2 ghi N số nguyên A_1, A_2, \dots, A_N ($|A_i| \leq 10^6$).
- Dòng thứ 3 ghi N số nguyên B_1, B_2, \dots, B_N ($|B_i| \leq 10^6$).

Kết quả ghi ra file **BalanceSequence.Out** là tổng S tính được.

Ví dụ:

BalanceSequence.Inp	BalanceSequence.Out
4	6
1 2 3 4	
4 4 4 4	

**10. Chữ số thuộc số - DigitInNum.Cpp**

Cho số tự nhiên N và chữ số k . Kiểm tra xem các chữ số của N có chữ số nào bằng k hay không?

Dữ liệu cho trong file **DigitInNum.Inp** gồm:

- Dòng đầu ghi số nguyên dương T ($T \leq 100$) là số cặp số: Số N và chữ số k .
- T dòng tiếp theo, mỗi dòng ghi hai số nguyên N và chữ số k ($N \leq 10^9$).

Kết quả ghi ra file **DigitInNum.Out** gồm T dòng là kết quả kiểm tra của T cặp số. Nếu chữ số k có trong N thì ghi ra “Yes”; Ngược lại thì ghi ra “No”.

Ví dụ:

DigitInNum.Inp	DigitInNum.Out
3	No
12 4	Yes
12 2	Yes
123 3	

**11. Cặp số tương thích - TuongThich.Cpp**

Cặp số gồm hai số nguyên a và b được gọi là tương thích nếu nó có ít nhất một chữ số chung.

Ví dụ: Các cặp $(1, 21)$; $(123, -33)$ là hai cặp số tương thích. Cặp $(123, 56)$ không phải là cặp số tương thích.

Yêu cầu: Cho hai số nguyên a, b . Kiểm tra xem hai số này có tương thích với nhau hay không?

Dữ liệu cho trong file **TuongThich.Inp** gồm:



- Dòng đầu ghi số nguyên dương N ($N \leq 100$) là số cặp số cần kiểm tra.
- N dòng tiếp theo, mỗi dòng ghi hai số nguyên a và b ($|a|, |b| \leq 10^9$).

Kết quả ghi ra file **TuongThich.Out** gồm N dòng là kết quả kiểm tra của N cặp số. Nếu cặp số là tương thích thì ghi ra “Yes”; Nếu cặp số không phải là tương thích thì ghi ra “No”.

Ví dụ:

TuongThich.Inp	TuongThich.Out
3	No
1 2	Yes
12 21	Yes
123 345	



12☀. Chia kẹo cho 3 anh em – DivideCandy3.Cpp

Hiện tại, 3 anh em Tôm, Tép, Cua có số kẹo lần lượt là a, b, c . Bố X viên kẹo. Tính xem, liệu Bố có thể cho 3 anh em X viên kẹo để số kẹo mỗi người sau khi cho là bằng nhau.

Dữ liệu cho trong file **DivideCandy3.Inp** gồm:

- Dòng đầu ghi 3 số nguyên dương a, b, c ($a, b, c \leq 100$) tương ứng là số kẹo của ba anh em Tôm, Tép, Cua.
- Dòng thứ 2 ghi số nguyên dương X là số kẹo của Bố sẽ cho cả ba anh em.

Kết quả ghi trong file **DivideCandy3.Out** là ‘Yes’ nếu có cách chia, ngược lại ghi ‘No’.

Ví dụ:

DivideCandy3.Inp	DivideCandy3.Out	Giải thích
3 4 5 3	Yes	Cho Tôm 2 cái: Tép 1 cái và Cua 0 cái.

**13☀. (Medium) Xóa số tính điểm – EraseMark1.Cpp**

Em Tép viết lên bảng một dãy số gồm $2N$ số nguyên không âm A_1, A_2, \dots, A_{2N} . Anh Tôm sẽ thực hiện N lần xóa, mỗi lần xóa, Tôm chọn 2 số ở 2 đầu dãy và xóa đi hai số đó. Nếu 2 số bị xóa bằng nhau thì điểm của lần xóa là tổng 2 số bị xóa, nếu 2 số bị xóa khác nhau thì lần xóa này không được điểm.

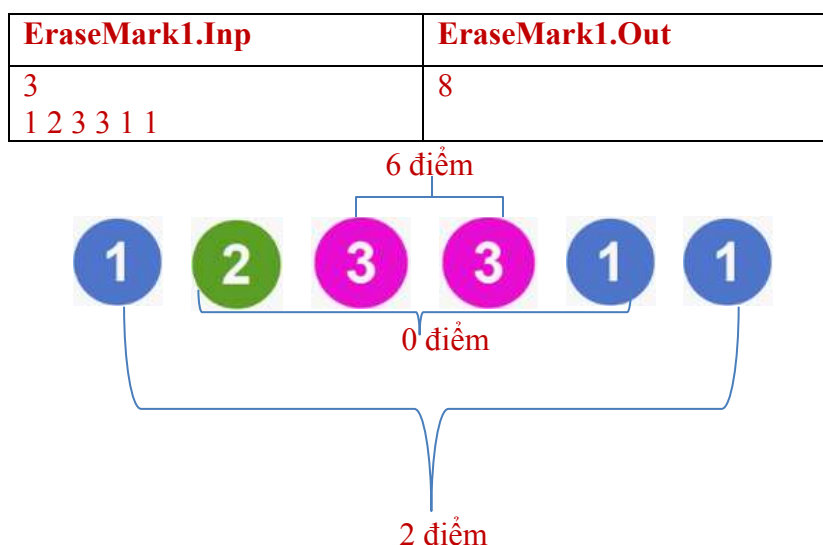
Yêu cầu: Cho dãy số mà em Tép đã viết lên bảng. Tính điểm nhận được của Anh Tôm.

Dữ liệu cho trong file **EraseMark1.Inp** gồm:

- Dòng đầu ghi số nguyên dương N ($N \leq 100000$).
- Dòng thứ 2 ghi $2N$ số nguyên không âm A_1, A_2, \dots, A_{2N} ($0 \leq A_i \leq 10^6$).

Kết quả ghi ra file **EraseMark1.Out** là số điểm mà anh Tôm khi xóa dãy nhận được.

Ví dụ:

**14☀. Xóa số trên đoạn và tính điểm – EraseMark2.Cpp**

Em Tép viết lên bảng một dãy số gồm $2N$ số nguyên không âm A_1, A_2, \dots, A_{2N} .

Anh Tôm sẽ chọn $2K$ số liên tiếp trong dãy gồm $2N$ số mà em Tép đã viết ($K \leq N$). Sau đó thực hiện K lần xóa (trên dãy gồm $2K$ số được chọn), mỗi lần xóa, Tôm chọn 2 số ở 2 đầu dãy và xóa đi hai số đó. Nếu 2 số bị xóa bằng nhau thì điểm của lần xóa là tổng 2 số bị xóa, nếu 2 số bị xóa khác nhau thì lần xóa này không được điểm.

Yêu cầu: Cho dãy số mà em Tép đã viết lên bảng. Tính xem, anh Tôm có thể nhận được số điểm lớn nhất bằng bao nhiêu.

Dữ liệu cho trong file **EraseMark2.Inp** gồm:

- Dòng đầu ghi 2 số nguyên dương N và K ($1 \leq K \leq N \leq 10000$; $K \leq 100$).
- Dòng thứ 2 ghi $2N$ số nguyên không âm A_1, A_2, \dots, A_{2N} ($0 \leq A_i \leq 10^6$).

Kết quả ghi ra file **EraseMark2.Out** là số điểm mà anh Tôm khi xóa dãy nhận được.

Ví dụ:

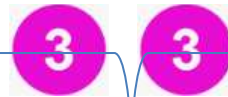


EraseMark2.Inp	EraseMark2.Out
3 1 1 2 3 3 1 1	6

Dãy ban đầu:



Tôm chọn 2 số liên tiếp:



Điểm nhận được: $3 + 3 = 6$.