

Chapter 3. Branching and Iteration

CẤU TRÚC if else

- *Dạng 1*

if (<biểu thức>
 <khối lệnh>;

Mô tả hoạt động:

- Nếu kết quả của <biểu thức> có giá trị khác 0 thì chương trình sẽ thực hiện <khối lệnh>;
- Nếu kết quả của <biểu thức> có giá trị bằng 0 thì chương trình sẽ bỏ qua <khối lệnh > để thực hiện câu lệnh tiếp theo sau if.

Dạng 2

```
if (<biểu thức >)  
    < khối lệnh 1>;  
else  
    < khối lệnh 2>;
```

Mô tả hoạt động

- Nếu kết quả của <biểu thức> là khác 0 thì chương trình sẽ thực hiện <khối lệnh 1>;
- Nếu kết quả của <biểu thức> là bằng 0 thì chương trình sẽ thực hiện <khối lệnh 2>.

Ví dụ 1

Giải phương trình dạng $ax + b = 0$

CẤU TRÚC switch . . . case

Cú pháp

```
switch ( <biểu thức> )
{
    case hằng_1 :    < khối lệnh 1>;
                    break;
    case hằng_2 :    < khối lệnh 2>;
                    break;
    ....
    case hằng_n :    < khối lệnh n>;
                    break;
    [ default :      < khối lệnh 0>;
                    break;]
}
```

Mô tả hoạt động

- Nếu giá trị của <biểu thức> trùng với hằng_i nào thì < khối lệnh i > tương ứng sẽ được thực hiện,
- Nếu sau < khối lệnh i > có lệnh break thì chương trình sẽ cho phép thoát ra khỏi cấu trúc switch case.
- Trong trường hợp < khối lệnh i > bằng rỗng thì < khối lệnh i+1> sẽ được xem xét tương tự.
- default là một thành phần không bắt buộc. Khi cấu trúc có sử dụng từ khóa default thì nghĩa là nếu giá trị của < biểu thức > không trùng với các giá trị từ hằng_1 đến hằng_n thì < khối lệnh 0 > sau default sẽ được thực hiện. Khi cấu trúc không có default thì sẽ thoát khỏi lệnh switch.
- Lệnh break có tác dụng ngắt chương trình, thoát khỏi < khối lệnh > chứa nó.

Ví dụ 2

Nhập hai giá trị tháng, năm của một ngày nào đó. Hãy cho biết tháng đó có bao nhiêu ngày.

Lưu ý: các tháng 1,3,5,7,8,10,12 có 31 ngày; các tháng 4,6,9,11 có 30 ngày.

Riêng tháng 2 thì có hai trường hợp sau:

Trường hợp 1: Nếu là năm nhuận thì tháng đó có 29 ngày (năm nhuận là năm chia hết cho 4 nhưng không chia hết cho 100 hoặc là năm chia hết cho 400),

Trường hợp hai: Nếu năm không nhuận thì tháng đó có 28 ngày.

Một số bộ test:



Test	Tháng/năm	Số ngày của tháng tương ứng
test 1	2/2008	29
test 2	2/2009	28
test 3	1/2009	31
test 4	11/2008	30
test 5	2/2000	28

```
1: #include <iostream>
2: using namespace std;
3: int main()
4: {
5:     int thang, songay, nam; // so ngay cua thang
6:     cin >> thang;
7:     switch (thang)
8:     {
9:     case 1:
10:    case 3:
11:    case 5:
12:    case 7:
13:    case 8:
14:    case 10:
15:    case 12:    songay=31;    break;
16:    case 4:
17:    case 6:
18:    case 9:
19:    case 11:    songay=30;    break;
20:    case 2:    cin >> nam;
21:    if (( nam %4 == 0 && nam %100 !=0 ) || nam % 400 == 0)
22:        songay = 29;
23:    else
24:        songay = 28;
25:    break;
26:    }
27:    cout<<songay;
28: }
```

Ví dụ 3

Viết chương trình nhập vào hai số nguyên a, b ; sau đó thực hiện một trong bốn phép tính cộng, trừ, nhân, chia (nếu gõ ký tự $+$ thì thực hiện phép cộng, nếu gõ ký tự $-$ thì thực hiện phép trừ,...).

```
1: #include <iostream>
2: using namespace std;
3: int main()
4: {
5:     long a,b;
6:     char pt;           // pt là kí?u k? t?
7:     cout << "\nChuong trinh thuc hien 4 phep toan so hoc\n";
8:     cout << "Nhap vao so a = "; cin >> a;
9:     cout << "Nhap vao so b = "; cin >> b;
10:    cout << "\nGo dau + de thuc hien phep cong:";
11:    cout << "\nGo dau - de thuc hien phep tru :";
12:    cout << "\nGo dau * de thuc hien phep tich:";
13:    cout << "\nGo dau / de thuc hien phep chia:";
14:    cout << "\nHay nhap vao lua chon:";
15:    cin >> pt;
16:    if ( pt == '+' )
17:        cout << "Ket qua la : " << a+b;
18:    else
19:        if (pt == '-' )
20:            cout << "Ket qua la : " << a-b;
21:        else
22:            if (pt == '*')
23:                cout << "Ket qua la : " << a*b;
24:            else
25:                if ( pt == '/')
26:                    cout << "Ket qua la : " << (float)a / b;
27:                else
28:                    cout << "Ban da chon khong dung phep toan!";
29: }
```

```
31: /*su dung cau truc switch
32: #include < iostream.h >
33: int main()
34: {
35:     long a,b;
36:     char pt;
37:     cout << "Chuong trinh thuc hien 4 phep toan so hoc\n";
38:     cout << "Nhap vao so a = ";      cin >> a;
39:     cout << "Nhap vao so b = ";      cin >> b;
40:     cout << "\nGo dau + de thuc hien phep cong:";
41:     cout << "\nGo dau - de thuc hien phep tru :";
42:     cout << "\nGo dau * de thuc hien phep tich:";
43:     cout << "\nGo dau / de thuc hien phep chia:";
44:     cout << "\nHay nhap vao lua chon:"; cin >> pt;
45:     switch (pt)
46:     {
47:     case '+':    cout<<"Ket qua la : "<< a+b;    break;
48:     case '-':    cout<<"Ket qua la : "<< a-b;    break;
49:     case '*':    cout<<"Ket qua la : "<< a*b;    break;
50:     case '/':    cout<<"Ket qua la : "<< (float) a / b; break;
51:     default :    cout<<"Ban da chon khong dung phep toan!";
52:     }
53: }
54: */
```

CẤU TRÚC for

```
for (<biểu thức 1>;<biểu thức điều kiện>; <biểu thức 2>)  
{  
    <khối lệnh> ;  
}
```

Mô tả hoạt động

- Tính <biểu thức 1>,
- Tính <biểu thức điều kiện>,
- Nếu giá trị của <biểu thức điều kiện> khác không (đúng) thì sẽ thực hiện <khối lệnh> rồi đến tính <biểu thức 2>.
- Sau khi thực hiện xong <biểu thức 2> thì quay trở lại tính giá trị của <biểu thức điều kiện> và bắt đầu một vòng lặp mới,... cứ thế cho đến khi giá trị của <biểu thức điều kiện> nhận giá trị bằng không (sai) thì cấu trúc lặp này sẽ kết thúc.

Nhận xét:

- Một trong các biểu thức: <biểu thức 1>; <biểu thức điều kiện>; <biểu thức 2> có thể không có (nhưng vẫn còn dấu ';' ngăn cách giữa chúng) thì câu lệnh vẫn hợp lệ.
- Nếu <biểu thức điều kiện> không có, hoặc là một hằng số khác không, thì vòng lặp for coi như lặp lại mãi mãi (chỉ có thể thoát ra bằng lệnh break hay return).

Ví dụ 4

Nhập vào một số nguyên dương $n \leq 10000000000$.
Hãy cho biết số lượng các ước số dương của n .

Một số bộ test:

	test 1	test 2	test 3	test 4
n	1	100	1001	10001
Kết quả	1	9	2	4

```
1 //
2 #include<iostream>
3 // #include<bits/stdc++.h>
4 using namespace std;
5 int main()
6 {
7     long n;
8     int d=1;
9     cout<<"Nhập vào số n = ";
10    cin>>n;
11    for (long i=1; i<=n/2; i++)
12        if (n % i==0)
13            d++;
14    cout<<"Số lượng ước số của số n : "<<d;
15    cout<<endl;
16    return 0;
17 }
18
```

Ví dụ 5

Nhập vào số tự nhiên n .

a. Kiểm tra n có phải là số hoàn chỉnh không ?

b. Kiểm tra n có phải là số chính phương không ?

c. Kiểm tra n có phải là số nguyên tố không ?

a'. Tìm các số hoàn chỉnh nhỏ hơn hoặc bằng n .

b'. Tìm các số chính phương nhỏ hơn hoặc bằng n .

c'. Tìm các số nguyên tố nhỏ hơn hoặc bằng n .

(yêu cầu bổ sung: Giải các câu a', b', c' bằng cách có sử dụng cấu trúc for và không sử dụng cấu trúc for).

```
#include<iostream.h>

int main()
{
    int n,s=0;
    cout<<"Nhap so n = ";cin>>n;
    for (int i=1;i<=n/2;i++)
        if (n%i==0)
            s=s+i;
    if (s==n)
        cout<<n<<" la so hoan hao";
    else
        cout<<n<<" khong la so hoan hao";
    cout<<endl;
    return 0;
}
```

```
#include<iostream.h>

int main()
{
    int n;
    cout<<"Nhap so n = ";cin>>n;
    for (int i=1;i<=n;i++)
    {
        int s=0;
        for (int j=1;j<=i/2;j++)
            if (i%j==0)
                s=s+j;
        if (s==i) cout<<i<<" ";
    }
    cout<<endl;
    return 0;
}
```

```
#include<iostream.h>
#include<math.h>
int main()
{
    int n;
    cout<<"Nhap so n = ";cin>>n;
    if (sqrt(n)==int(sqrt(n)))
        cout<<n<<" la so cp";
    else
        cout<<n<<" khong la cp";
    cout<<endl;
    return 0;
}
```

```
#include<iostream.h>
#include<math.h>
int main()
{
    int n;
    cout<<"Nhap so n = ";cin>>n;
    for (int i=1;i<=n;i++)
        if (sqrt(i)==int(sqrt(i)))
            cout<<i<<" ";
    cout<<endl;
    return 0;
}
```

```
#include<iostream.h>

int main()
{
    int n,d=0;

    cout<<"Nhap so n = ";cin>>n;

    for (int i=1;i<=n;i++)
        if (n%i==0) d++;

    if (d==2)
        cout<<n<<" la so nguyen to";

    else
        cout<<n<<" khong la so nguyen to";

    cout<<endl;

    return 0;
}
```

```
#include<iostream.h>

int main()
{
    int n;

    cout<<"Nhap so n = ";cin>>n;

    for (int i=1;i<=n;i++)
    {
        int d=0;

        for (int j=1;j<=i;j++)
            if (i%j==0) d++;

        if (d==2)
            cout<<i<<" ";

    }

    cout<<endl;

    return 0;
}
```

CẤU TRÚC while

```
while (< biểu thức >)  
{  
    <khối lệnh>;  
}
```


Mô tả hoạt động

- Trong cấu trúc của lệnh này, trước tiên sẽ tính giá trị của < **biểu thức** >.
- Nếu giá trị của < **biểu thức** > khác không (đúng) thì sẽ thực hiện <**khối lệnh**>, sau đó quay trở lại tính giá trị của < **biểu thức** >.
- Vòng lặp **while** này sẽ kết thúc việc lặp khi giá trị của < **biểu thức** > nhận giá trị không (sai).

Nhận xét:

- < biểu thức > trong vòng lặp while có thể là một biểu thức bất kỳ, ngôn ngữ lập trình C chỉ quan tâm rằng giá trị của < biểu thức > là bằng không hay khác không.
- Khi giá trị của < biểu thức > còn khác không thì còn làm. Khi giá trị của < biểu thức > bằng không thì thoát khỏi vòng lặp.
- <khối lệnh> trong thân vòng lặp while có thể không được thực hiện lần nào.

Ví dụ 6

Viết chương trình nhập vào một số nguyên dương n ($0 < n < 10000000000$). Hãy tính tổng các chữ số của số đó.

Một số bộ test:

test	test 1	test 2	test 3	test 4
n	1	2010	999111	1010101
Kết quả	1	3	30	4

```
1
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6  long n;
7  cout<<"Nhập vào số n = ";cin>>n;
8  int s=0;
9  while (n>0)
10 {
11     s=s+n%10;
12     n=n/10;
13 }
14 cout<<s;
15 cout<<endl;
16 return 0;
17 }
18
```

Ghi chú: Hai câu lệnh sau tương đương

```
for ( <biểu thức 1>; <biểu thức  
điều kiện>; <biểu thức 2> )  
{  
    <khối lệnh> ;  
}
```

```
<biểu thức 1>;  
while (<biểu thức điều kiện > )  
{  
    <khối lệnh >;  
    <biểu thức 2>;  
}
```

CẤU TRÚC do while

do

{

< khối lệnh >;

}

while < biểu thức >;

Mô tả hoạt động

- Trước hết < **khối lệnh** > sẽ được thực hiện, sau đó tính giá trị của < **biểu thức** >.
- Nếu giá trị của < **biểu thức** > khác không (đúng) thì quay trở lại thực hiện < **khối lệnh** > sau đó lại tính giá trị của < **biểu thức** >.
- Vòng lặp này kết thúc khi giá trị của < **biểu thức** > nhận giá trị không (sai).

Câu lệnh **break**

- Khi có nhiều vòng lặp lồng nhau, câu lệnh **break** sẽ thoát ra khỏi vòng lặp (hoặc lệnh **switch**) bên trong nhất chứa nó.

Ví dụ 7

Sinh viên tham khảo việc sử dụng các lệnh *break*, *continue* qua bài toán sau:
Nhập vào số tự nhiên n . Hãy **tìm** các số nguyên tố $\leq n$.

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int n;
6      cin>>n;
7      cout<<"Nhập vào số n = ";
8      for (int i=2; i<=n; i++)
9      {
10         int j;
11         for (j=2; j<i; j++)
12             if (i%j==0) break;
13         if (j<i) continue;
14         else cout<<i<<" ";
15     }
16     cout<<endl;
17     return 0;
18 }
```

Câu lệnh continue

- Trái với câu lệnh **break**, câu lệnh **continue**, chỉ được dùng trong vòng lặp, dùng để bắt đầu một lần lặp mới của vòng lặp bên trong nhất chứa nó.

Bài tập

BT1.

Nhập vào ba số dương a, b, c . Hỏi a, b, c có phải số đo ba cạnh của một tam giác hay không ? Nếu có, hãy xét xem đó là tam giác đều, tam giác vuông, tam giác cân hay tam giác thường ?

BT2.

Mô phỏng trò chơi búa, bao, kéo. Người 1: ra b,a,k ứng với búa, bao, kéo. Người 2: ra b,a,k ứng với búa, bao, kéo. Chương trình trả lời Người 1 thắng hay Người 2 thắng hay hòa nhau (SV tìm hiểu về kiểu ký tự).

BT3.

a.Tìm ước số chung lớn nhất của 2 số tự nhiên a, b .

b.Tìm bội số chung nhỏ nhất của 2 số tự nhiên a, b .

c.Tối giản phân số a/b

d.Tính tổng 2 phân số $(a/b + c/d)$. Yêu cầu phân số kết quả ở dạng tối giản.

(giả thiết a, b, c, d là các số nguyên dương)

BT4.

Nhập số tự nhiên n .

a.Xuất các chữ số của số n theo chiều ngược lại.

b.Kiểm tra n có phải là số đối xứng hay không ?

BT5.

- a. Nhập vào số tự nhiên n . Hãy kiểm tra xem n có phải là số hoàn chỉnh hay không (n là số hoàn chỉnh nếu nó bằng tổng các ước số thực sự của nó). Ví dụ 6 là số hoàn chỉnh vì $6=1+2+3$.
- b. Nhập vào số tự nhiên n . Kiểm tra xem n có phải là số chính phương hay không ? Ví dụ 1,4,9,16 là các số chính phương.
- c. Nhập vào số tự nhiên n . Kiểm tra xem n có phải là số nguyên tố hay không. Ví dụ 2,3,5,7 là các số nguyên tố.
- d. Tương tự, kiểm tra số armstrong
- e. Tương tự, kiểm tra số đối xứng (Ví dụ: 12221,101,4,...)
- f. Tương tự, tìm số viết ngược (Ví dụ: 1423->3241)

BT6.

Nhập vào số nguyên dương n . Hãy thực hiện các công việc sau:

- a. Đếm số lượng số nguyên tố nhỏ hơn hoặc bằng n .
- b. Tìm số nguyên tố thứ n (dãy các số nguyên tố: 2,3,5,7,11,...)
- c. Tìm số nguyên tố gần n nhất (có thể lớn hơn hoặc nhỏ hơn n).
- d. Tìm n số nguyên tố đầu tiên.

Ví dụ: $n=8$ thì kết quả là:

- a. 4
- b. 19
- c. 7
- d. 2 3 5 7 11 13 17 19

BT7.

Tính tổng $S=1 + 1.2 + 1.2.3 +... + 1.2.3.4....n$ (với n là số nguyên dương).

BT8.

Cho n là số nguyên dương.

Hãy tính tổng $S(n)=1+\frac{2}{1+\frac{1}{2}}+\frac{3}{1+\frac{1}{2}+\frac{1}{3}}+\dots+\frac{n}{1+\frac{1}{2}+\frac{1}{3}+\dots+\frac{1}{n}}$

BT9.

Giả sử $n \geq 1$ và x là số thực. Hãy viết chương trình tính giá trị của biểu thức sau:

$$S(n, x) = \frac{x}{1} - \frac{x^2}{1+\frac{1}{2}} + \frac{x^3}{1+\frac{1}{2}+\frac{1}{3}} - \dots + (-1)^{n-1} \frac{x^n}{1+\frac{1}{2}+\dots+\frac{1}{n}}$$

BT10.

Liệt kê các số tự nhiên k thỏa đồng thời ba điều kiện sau:

- k là số có 5 chữ số;
- k là số nguyên tố;
- k là số đối xứng (k là số đối xứng nếu đọc xuôi hay đọc ngược các chữ số của k ta đều nhận được một số như nhau. Ví dụ số: 10001 là số đối xứng).

BT11.

a.Tìm ngày kế sau và ngày kế trước của ngày $d/m/y$.

b.Từ ngày 1/1/1 đến ngày $d/m/y$ có bao nhiêu ngày ? Với d,m,y nhập từ bàn phím.

c.Đổi năm dương lịch ra năm âm lịch

d.Đổi năm âm lịch ra năm dương lịch

BT12.

Nhập vào số báo danh, họ tên, điểm thi 3 môn Toán học, Vật lý, Hóa hóa.

Xuất tổng điểm 3 môn thi trên. Yêu cầu làm tròn đến 0.5

Ví dụ 19.35 làm tròn 19.50; 19.80 làm tròn 20.00

*SV tìm hiểu lệnh nhập chuỗi gets

BT13.

Trong mặt phẳng tọa độ OXY cho đa giác P có n đỉnh $A_1A_2...A_n$; tọa độ các đỉnh lần lượt là x_i, y_i là các số nguyên, với $i=1..n$ (P là đa giác lồi hoặc lõm). Hãy trình bày thuật toán tìm diện tích đa giác P .

Ví dụ: cho $n=8$ và tọa độ các đỉnh $A_1(1,-1)$, $A_2(4,-1)$, $A_3(5,1)$, $A_4(5,2)$, $A_5(3,1)$, $A_6(3,4)$, $A_7(0,2)$, $A_8(2,1)$ thì kết quả là 12.

$|\sum_{i=1}^n (x[i+1] - x[i])(y[i+1] + y[i])/2|$; trong đó tọa độ đỉnh $n+1$ trùng đỉnh 1.