

NHẬP MÔN LẬP TRÌNH

CHƯƠNG 3. CÁC CẤU TRÚC ĐIỀU KHIỂN CHƯƠNG TRÌNH

I. Cấu trúc lệnh rẽ nhánh

Nội dung

1. Lệnh rẽ nhánh

1. Lệnh if
2. Lệnh switch-case
3. Bài tập

2. Lệnh lặp

1. Lệnh for
2. Lệnh while
3. Lệnh do – while
4. Bài tập

1. Lệnh if

Câu lệnh if cho phép thay đổi luồng thực thi của câu lệnh dựa vào điều kiện, một câu lệnh hoặc một khối các câu lệnh sẽ được quyết định thực thi hay không được thực thi.

Ví dụ: Xét tính chẵn lẻ của một số nguyên a .

⇒ Chọn 1 trong 2 kết quả:

- Số a là số chẵn nếu a chia hết cho 2
- Số a là số lẻ nếu a không chia hết cho 2

1. Lệnh if (tt)

(2)

- ☐ Lệnh if đơn giản
- ☐ Lệnh if-else đơn giản
- ☐ Lệnh if lồng nhau

Lệnh if đơn giản

□ Cú pháp

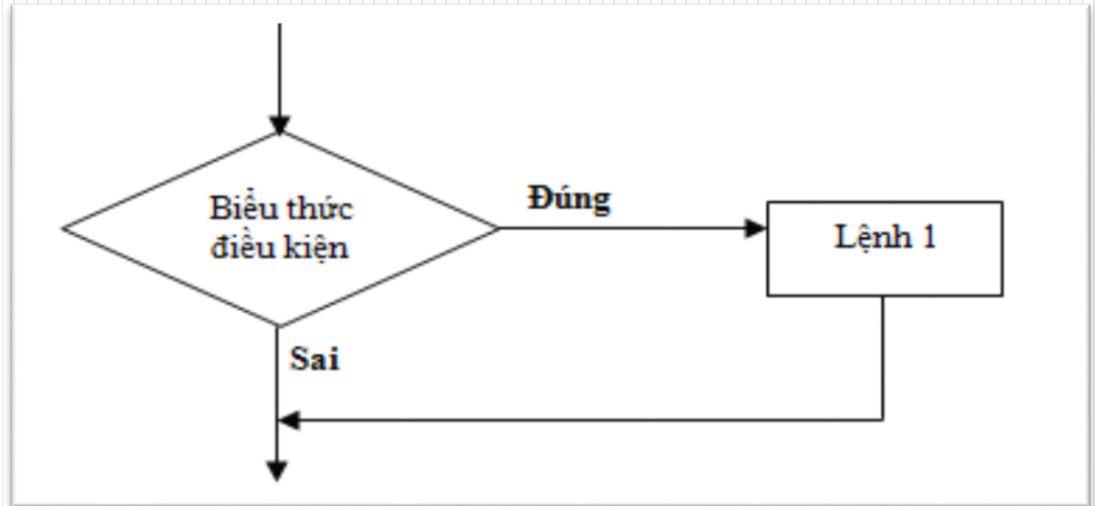
```
if (<Biểu thức điều kiện>)  
    <Lệnh 1>
```

Lệnh if đơn giản (tt)

(2)

□ Lưu đồ

**if (<BTĐK>
<Lệnh 1>**



Lệnh 1: một câu lệnh đơn / một khối lệnh / một câu lệnh phức.

Các bước thực hiện:

B1: tính giá trị biểu thức điều kiện .

B2: kiểm tra

- Nếu điều kiện đúng (bằng 1) thì thực hiện Lệnh 1
- Ngược lại, nếu điều kiện sai (bằng 0) thì bỏ qua Lệnh 1

Lệnh if đơn giản (tt)

(3)

- ❑ Ví dụ: Thực hiện chương trình nhập vào một số thực a. In ra màn hình kết quả nghịch đảo của a khi $a \neq 0$

```
float a;  
printf("Nhap a = "); scanf("%f",&a);  
if (a !=0 )  
    printf("Nghich dao cua %f la %f",a,1/a);  
printf(" \n Ket thuc chuong trinh");  
getch();  
return 0;
```

- ❑ Chú ý: lệnh *printf(" \n Ket thuc chuong trinh");* được thực hiện bình thường cho dù giá trị biểu thức điều kiện trong if đúng hay sai vì nó không phụ thuộc vào lệnh if

Lệnh if-else đơn giản

□ Cú pháp

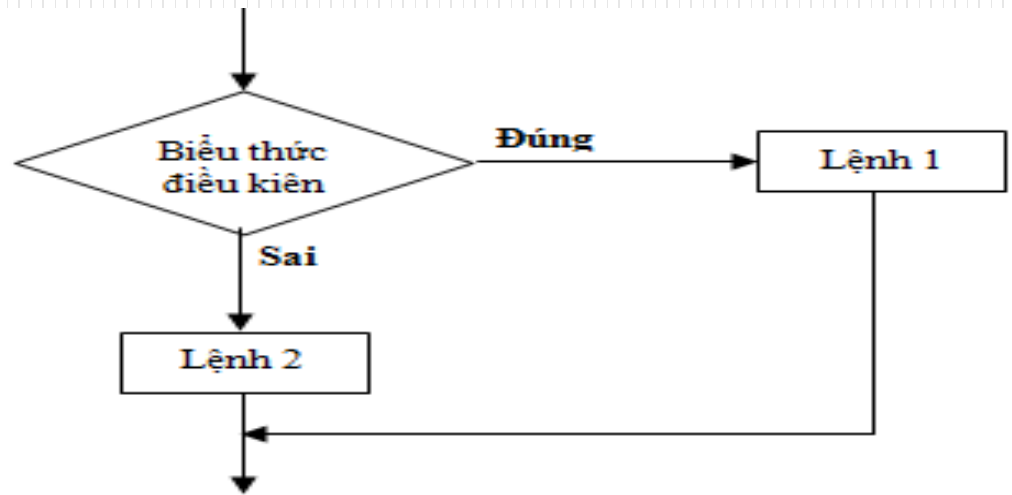
```
if (<Biểu thức điều kiện>
    <Lệnh 1>
else
    <Lệnh 2>
```

Lệnh if-else đơn giản (tt)

(2)

□ Lưu đồ

```
if (<BTDK>)  
    <Lệnh 1>  
else  
    <Lệnh 2>
```



Lệnh 1, Lệnh 2: một câu lệnh đơn / một khối lệnh / một câu lệnh phức.

Các bước thực hiện:

B1: tính giá trị biểu thức điều kiện .

B2: kiểm tra

- Nếu điều kiện đúng (bằng 1) thì thực hiện Lệnh 1
- Ngược lại, nếu điều kiện sai (bằng 0) thì thực hiện Lệnh 2

Lệnh if-else đơn giản (tt)

(3)

- ❑ Ví dụ: Viết chương trình nhập vào một số thực a. In ra màn hình kết quả nghịch đảo của a khi $a \neq 0$, khi $a=0$ in ra thông báo “Khong the tim duoc nghich dao cua a”

```
float a;  
printf("Nhap a = "); scanf("%f",&a);  
if (a !=0 )  
    printf("Nghich dao cua %f la %f",a,1/a);  
else  
    printf("Khong the tim nghich dao cua a);  
printf(" \n Ket thuc chuong trinh");
```

- ❑ Chú ý: lệnh *printf(" \n Ket thuc chuong trinh");* được thực hiện bình thường cho dù giá trị biểu thức điều kiện trong if đúng hay sai vì nó không phụ thuộc vào lệnh if-else

Lệnh if lồng nhau

- Câu lệnh if đơn giản và if-else có thể lồng vào nhau
- Ví dụ: Nhập vào giá trị 2 số nguyên a, b. So sánh số a với số b vừa nhập

```
int a, b;  
printf("Nhap a = "); scanf("%d",&a);  
printf("Nhap b = "); scanf("%d",&b);  
if (a > b )  
    printf("a lon hon b");  
else  
    if(a==b)  
        printf("hai so bang nhau");  
    else  
        printf("a nho hon b");
```

2. Lệnh switch-case

Câu lệnh rẽ nhánh switch-case cho phép lựa chọn một trong các lựa chọn đã đưa ra. Nếu biểu thức có giá trị bằng với một giá trị sau từ khóa case nào đó, thì câu lệnh tương ứng sẽ được thực thi.

Ví dụ: Viết chương trình nhập vào giá trị tháng của năm, xuất số ngày trong tháng vừa nhập.

2. Lệnh switch-case (tt)

(2)

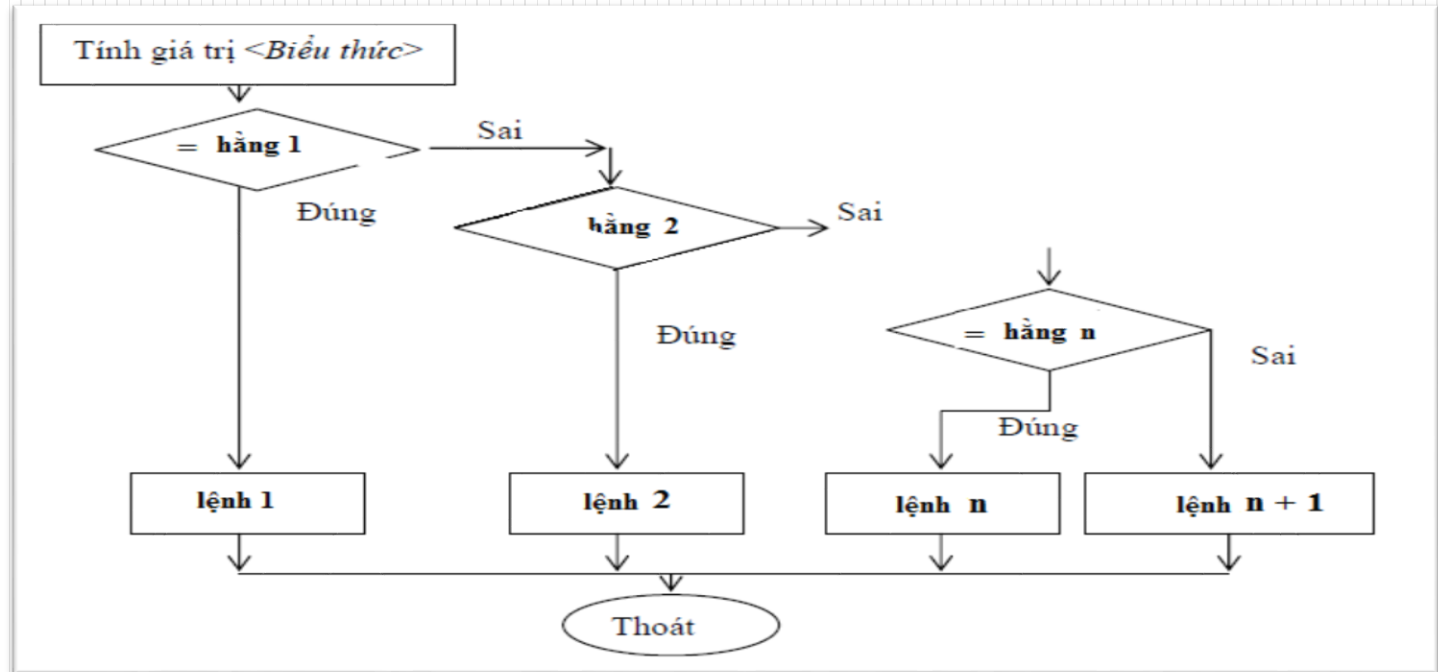
□ Cú pháp:

```
switch (<biểu thức>
{
    case <hằng 1> : lệnh 1 ;
    break;
    case <hằng 2> : lệnh 2 ;
    break;
    .....
    case <hằng n> : lệnh n ;
    break ;
    default      :      lệnh n+1;
}
```

2. Lệnh switch-case (tt)

(3)

□ Lưu đồ:



B1: Tính giá trị của biểu thức trước.

B2: - Nếu giá trị của biểu thức bằng hằng 1 thì thực hiện lệnh 1 rồi thoát.

- Nếu giá trị của biểu thức khác hằng 1 thì so sánh với hằng 2, nếu bằng hằng 2 thì thực hiện lệnh 2 rồi thoát.

- Cứ như thế, so sánh tới hằng n.

- Nếu tất cả các phép so sánh trên đều sai thì thực hiện lệnh n+1 mặc định của trường hợp *default*.

2. Lệnh switch-case (tt)

(4)

- ❑ Ví dụ: Viết chương trình nhập vào giá trị tháng của năm, xuất số ngày trong tháng vừa nhập.

Bài tập (tt)

1. Viết chương trình nhập vào hai số a, b. Kiểm tra a có là bội số của b không.
2. Viết chương trình nhập vào đơn giá một mặt hàng, và số lượng bán của mặt hàng. Tính tiền khách phải trả, với thông tin như sau:
Thành tiền: đơn giá * số lượng.
Giảm giá : Nếu thành tiền > 100, thì giảm 3% thành tiền, ngược lại không giảm.
Tổng tiền phải trả: thành tiền – giảm giá.
3. Viết chương trình hỗ trợ cách giải phương trình bậc 1 ($ax + b = 0$)
4. Nhập vào 1 số bất kỳ (0->9), cho biết cách đọc số vừa nhập.
5. Tính chu vi, diện tích của một trong các hình bên dưới (người dùng chọn hình): Hình vuông, Hình chữ nhật, Hình tròn.
6. Viết chương trình nhập vào hai số nguyên và lựa chọn phép toán (cộng, trừ, nhân, chia) để tính kết quả.

II. Cấu trúc lệnh lặp

1. Khái niệm

Lệnh lặp là một câu lệnh, một đoạn lệnh trong chương trình được thực hiện lặp đi lặp lại nhiều lần cho đến khi một điều kiện xác định được thỏa mãn.

Một lệnh lặp cho phép lặp lại các câu lệnh nhiều lần.

Ví dụ: Viết chương trình xuất câu Xin chào 100 lần.

2. Lệnh for

- ❑ Lệnh for thực thi việc lặp lại một câu lệnh, một khối lệnh nhiều lần với **số lần lặp xác định trước**.
- ❑ Cú pháp

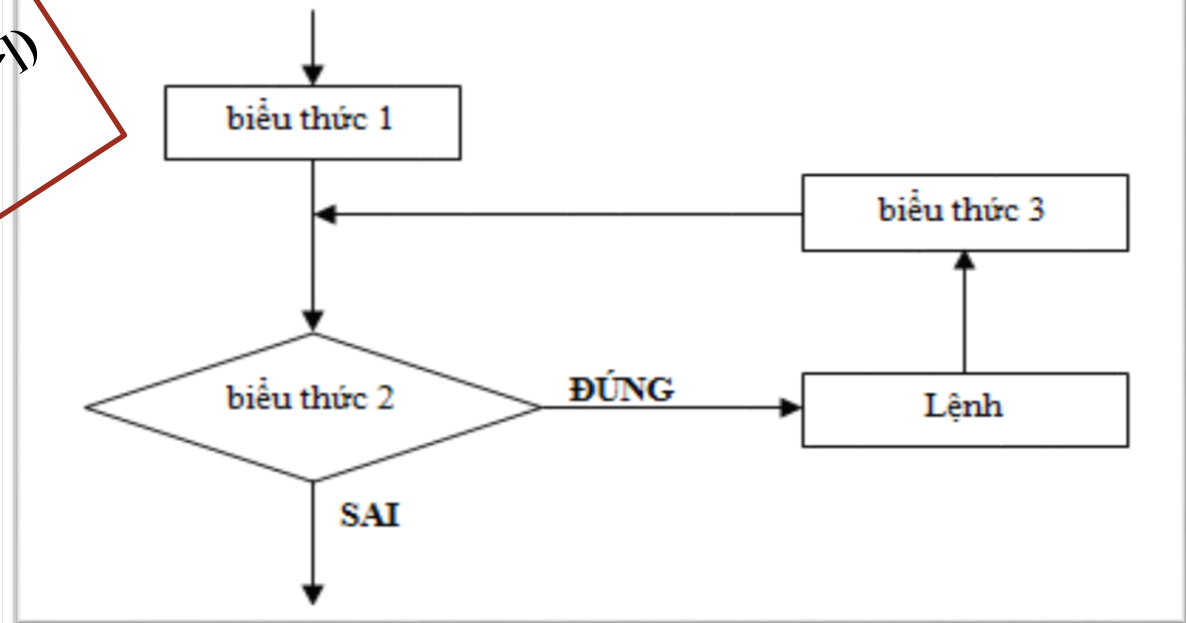
```
for ( [ <biểu thức 1>]; [ <biểu thức 2>] ; [ <biểu thức 3>])  
    <câu lệnh>;
```

2.Lệnh for (tt)

(2)

□ Lưu đồ

for (l < BT1 > l ; l < BT2 > l ; l < BT3 > l)
 < câu lệnh >;



Bước 1: Xác định giá trị của biểu thức 1.

Bước 2: Xác định giá trị của biểu thức 2.

Bước 3: Nếu biểu thức 2 sai thì sẽ thoát vòng lặp for;

Nếu biểu thức 2 đúng thì máy sẽ thực hiện Lệnh.

Bước 4: Tính giá trị của biểu thức 3 và quay lại Bước 2.

2.Lệnh for (tt)

(3)

- ❑ Ví dụ: Viết chương trình xuất câu Xin chào lên màn hình 20 lần.

```
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");  
printf("\n Xin chao!!! ");
```



```
int i;  
for (i=1; i<=20; i++)  
    printf("\n Xin chao!!! ");
```

2.Lệnh for (tt)

(4)

- ❑ Ví dụ: Viết chương trình nhập vào một số nguyên n. Tính tổng của các số nguyên từ 1 đến n.

```
int n,i,tong;  
printf("\n Nhap vao so nguyen duong n:");  
scanf("%d",&n);  
tong=0;  
for (i=1; i<=n; i++)  
    tong+=i;  
printf("\n Tong tu 1 den %d =%d ",n,tong);
```

2.Lệnh for (tt)

(5)

❑ Chú ý:

- ❑ Biểu thức 1 là biểu thức gán trị khởi động cho biến lặp.
- ❑ Biểu thức 2 là biểu thức điều kiện. Nếu biểu thức 2 vắng mặt, điều kiện luôn đúng.
- ❑ Biểu thức 3 thông thường là biểu thức thay đổi điều kiện.
- ❑ Biểu thức 1, 3 có thể gồm nhiều biểu thức cách nhau bởi dấu phẩy.
- ❑ Biểu thức thứ 2 có thể bao gồm nhiều biểu thức, nhưng tính đúng sai của nó được xem là tính đúng sai của biểu thức cuối cùng.
- ❑ Các vòng for có thể lồng với nhau để thực hiện một câu lệnh nào đó.

2.Lệnh for (tt)

(6)

Ví dụ: Viết chương trình xuất bảng cửu chương từ 1- 9

```
for( i = 1 ; i <= 9 ; i++)  
{  
    printf(“\n bang cuu chuong thu %d ”, i);  
    for(j = 1 ; j <= 9 ; j ++)  
        printf(“\n  %d x %d = %d ”, i, j, i * j);  
}
```

3. Lệnh while

- ❑ Lệnh while thực thi việc lặp lại một khối lệnh khi điều kiện kiểm tra là đúng. Điều kiện sẽ được kiểm tra trước khi vào thân vòng lặp do đó nếu có thay đổi giá trị kiểm tra ở trong thân vòng lặp thì khối lệnh vẫn được thực thi cho đến khi kết thúc khối lệnh.
- ❑ Cú pháp

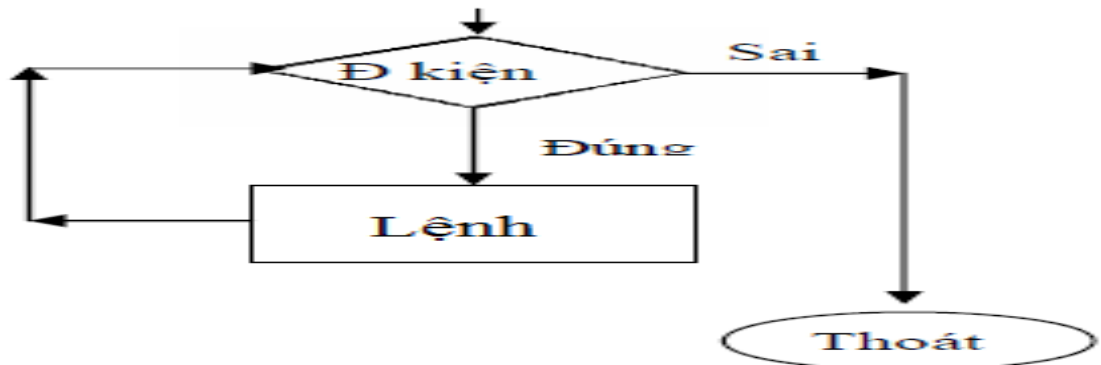
```
while(<biểu thức điều kiện>)  
< Lệnh >
```

3.Lệnh while (tt)

(2)

❑ Lưu đồ

while(<BTĐK>)
< Lệnh >



Bước 1: Tính giá trị của biểu thức điều kiện.

Bước 2: Nếu biểu thức điều kiện là sai (0), thì máy sẽ thoát ra khỏi vòng lặp.

Nếu biểu thức điều kiện là đúng (1) thì máy sẽ thực hiện câu lệnh và quay lại bước 1.

❑ Chú ý: Trong biểu thức điều kiện có thể gồm nhiều biểu thức cách nhau bởi dấu phẩy “,” nhưng tính đúng sai của nó là tính đúng sai của biểu thức sau cùng.

3.Lệnh while (tt)

(3)

- ❑ Ví dụ: Viết chương trình in các số nguyên từ 1 đến n, trong đó n nhập từ phím

```
int i,n;  
printf("\n Nhap n:"); scanf("%d", &n);  
printf("\n Day so tu 1 den %d :",n);  
i=1;  
while (i<=n)  
    printf("%d ",i++);
```

3.Lệnh while (tt)

(4)

❑ Chú ý:

- ❑ Giá trị của biến được sử dụng trong biểu thức phải được thiết lập trước khi vòng lặp **while** thực hiện. Đây gọi là bước khởi tạo giá trị. Lệnh này chỉ thực hiện một lần trước khi thực hiện vòng lặp.
- ❑ Thân vòng lặp phải làm thay đổi giá trị của biến trong biểu thức kiểm tra. Biến này được gọi là biến tăng (**incremented**) nếu giá trị trong thân vòng lặp tăng, và được gọi là biến giảm (**decremented**) nếu giá trị giảm.

4. Lệnh do-while

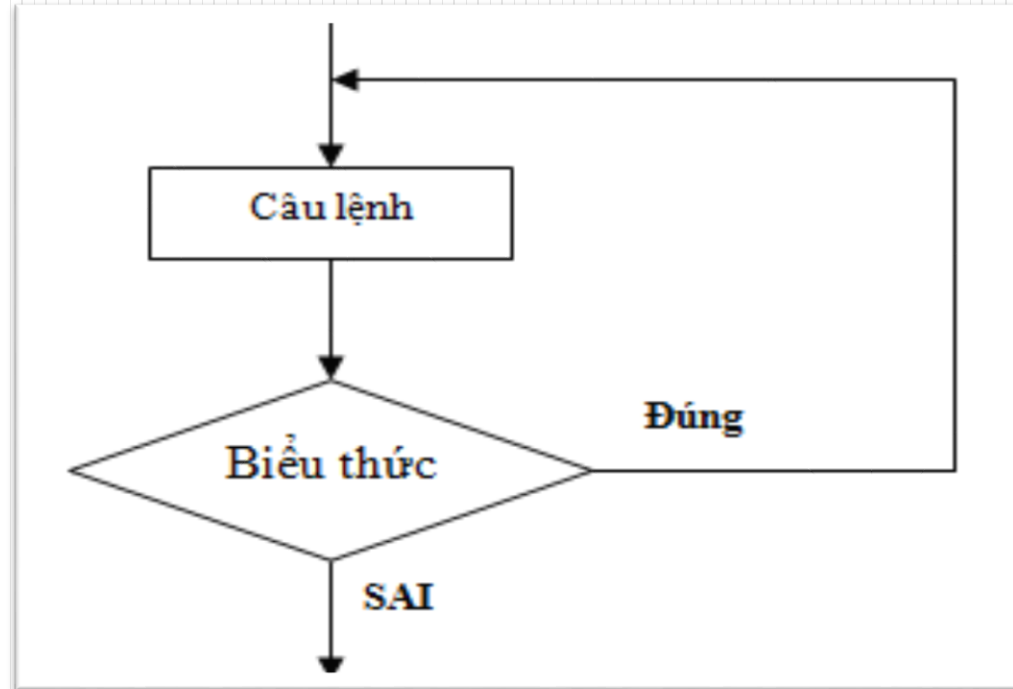
- ❑ Lệnh do...while thực thi lặp lại một khối lệnh nhiều lần.
- ❑ Khối lệnh được thực hiện ít nhất một lần. Sau đó sẽ kiểm tra điều kiện nếu điều kiện là đúng thì tiếp tục thực thi khối lệnh. Nếu điều kiện là sai thì kết thúc vòng lặp.
- ❑ Cú pháp

```
do  
    < câu lệnh >  
while(<biểu thức>)
```

4. Lệnh do-while (tt)

(2)

□ Lưu đồ:



Bước 1: Câu lệnh được thực hiện trước tiên.

Bước 2: Tính giá trị của biểu thức, nếu biểu thức đúng thì quay lại bước 1, nếu giá trị biểu thức sai thì ngừng vòng lặp.

4. Lệnh do-while (tt)

(3)

- ❑ Ví dụ: Viết chương trình nhập vào một số dương, nếu nhập số âm yêu cầu nhập lại.

```
int value;  
do  
{  
    printf( "Nhap vao mot so duong");  
    scanf("%d",&value);  
} while(value <=0);
```

5. Lệnh break

- Câu lệnh break được dùng để thoát khỏi vòng lặp (bất chấp biểu thức điều kiện trong câu lệnh lặp đúng)

```
int i, j ;
printf(“ nhập hai số nguyên dương i và j :”);
scanf(“%d%d”,&i,&j);
for( ; i > 0 && j > 0; i- -,j- -)
{
    if( j == 5 ) break;
    printf(“ i = %d, j = %d ”, i, j);
}
```

6. Lệnh continue

- Câu lệnh continue được dùng để bỏ qua 1 bước lặp hiện tại của vòng lặp.

```
int i, tong ;  
for( i=1; i < 20; i++)  
{  
    if( i % 2 ==0 ) continue;  
    tong = tong + i;  
}  
printf(“ ket qua: tong= %d ”, tong);
```

Bài tập

1. Viết chương trình in trên màn hình các số từ 1->100, các số ngăn cách nhau bởi 1 đoạn khoảng trắng.
2. Viết chương trình vẽ hình chữ nhật có kích thước $d \times r$, trong đó d là chiều dài, và r là chiều rộng được nhập từ phím.

```
*      *      *      *      *  
*      *      *      *      *  
*      *      *      *      *
```

3. Viết chương trình hiển thị tất cả các số lẻ nhỏ hơn n , trong đó n nhập từ phím.

Bài tập (tt)

6. Viết chương trình in ra các số là bội số của 5 nhỏ hơn n , trong đó n nhập từ phím.
7. Viết chương trình nhập vào 2 số nguyên, tìm USCLN, BSCNN.
8. Viết chương trình đếm số lượng số chẵn trong $[n, m]$, trong đó n, m nhập từ phím.
9. Viết chương trình tính tổng các số tự nhiên nhỏ hơn n .
10. Viết chương trình tìm tổng các số tự nhiên lớn nhất nhỏ hơn 100.

