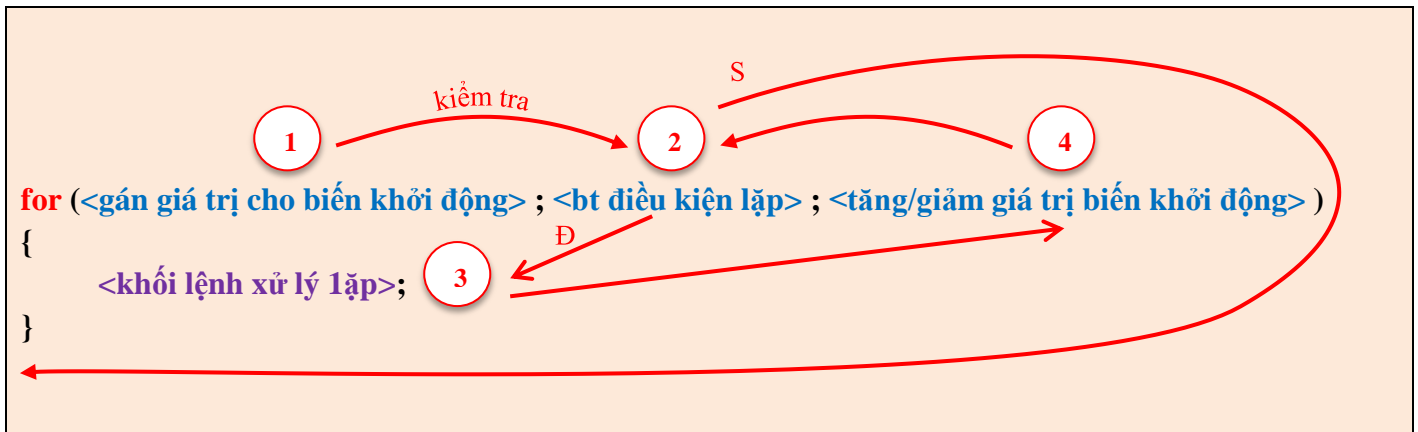


## NỘI DUNG CHÍNH BUỔI 6

### CÁC CÂU LỆNH LẶP

#### (1) LỆNH FOR

- Cú pháp:



a) Trong đó:

- <khối lệnh xử lý lặp> có thể gồm nhiều lệnh. Trường hợp chỉ có 1 lệnh xử lý lặp thì có thể bỏ cặp dấu { }.

**Ví dụ 1:** Viết đoạn CT tính  $S = 1 + 2 + 3 + \dots + n$  (với  $n > 0$ )

```
...
int S=0, n, i;
//Nhập liệu n>0
for(i=1; i<=n; i=i+1)
    S=S+i;
printf("S = %d", S);
...
```

⇒ **Debug (F10) CT và xem kết quả.**

#### ❖ Bổ sung lý thuyết Chương 2 – Trang 17

- **Toán tử 1 ngôi:** là toán tử chỉ tác động lên một toán hạng:

**++** (tăng 1 đơn vị) **--** (giảm 1 đơn vị)

#### ▪ Chú ý:

- Nếu toán tử đặt trước toán hạng thì toán hạng được tăng/giảm trước khi được dùng.
- Nếu toán tử đặt sau toán hạng thì toán hạng được dùng trước khi tăng/giảm.
- VD:

```
int x, y;
x = 10;
y = x++ + 5; //<=> y=x+5, x=x+1 => y=15, x=11
x = 10;
y = ++x + 5; //<=> x=x+1, y=x+5 => x=11, y=16
```

- **Toán tử 2 ngôi:** là toán tử tác động lên hai toán hạng:

**+ - \* / %** (chia lấy phần dư) **+= -= \*= /= %=**

- VD1:

```
int a = 9, b = 2, x, y;
float c = 9.0, z;
x = a/b; // x = 4 (chia nguyên)
y = a%b; // y = 1
z = c/b; // z = 4.5 (chia thực)
```

- VD2:

```
int n = 10;
n += 10; // n = n + 10 => n = 20
n -= 5; // n = n - 5 => n = 15
n *= 2; // n = n * 2 => n = 30
n /= 4; // n = n / 4 => n = 7
n %= 2; // n = n % 2 => n = 1
```

**Viết lại ví dụ 1:** Viết đoạn CT tính  $S = 1 + 2 + 3 + \dots + n$  (với  $n > 0$ )

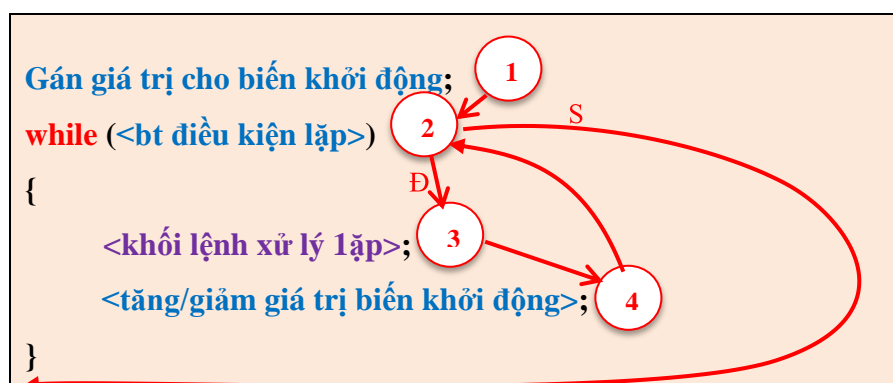
```
...
int S=0, n, i;
//Nhập liệu n>0
for (i=1; i<=n; i++)
    S+=i;
printf("S = %d", S);
...
```

**Ví dụ 2:** Viết đoạn CT tính  $n!$  như sau:  $T = 1 * 2 * 3 * \dots * n$  (với  $n > 0$ )

```
...
int T=1, n, i;
//Nhập liệu n>0
for (i=1; i<=n; i++)
    T*=i;
printf("%d! = %d", n, T);
...
```

## (2) LỆNH WHILE

- Cú pháp:



**Viết lại ví dụ 1:** Viết đoạn CT tính  $S = 1 + 2 + 3 + \dots + n$  (với  $n > 0$ )

```

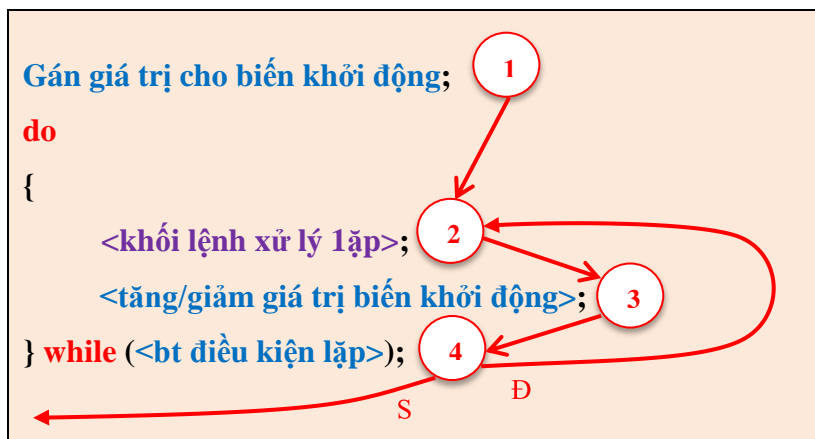
...
int S=0, n, i=1;
//Nhập liệu n>0
while (i<=n)
{
    S+=i;
    i++;
}
printf("S = %d", S);
...

```

⇒ **Debug (F10) CT và xem kết quả.**

**(3) LỆNH DO...WHILE...**

- Cú pháp:

**Viết lại ví dụ 1:** Viết đoạn CT tính  $S = 1 + 2 + 3 + \dots + n$  (với  $n > 0$ )

```

...
int S=0, n, i=1;
//Nhập liệu n>0
do
{
    S+=i;
    i++;
}while (i<=n);
printf("S = %d", S);
...

```

⇒ **Debug (F10) CT và xem kết quả.**

- Khác nhau giữa lệnh **while** và **do...while...**?

Lệnh <b>while</b>	Lệnh <b>do...while...</b>
<ul style="list-style-type: none"> <li>- Kiểm tra &lt;biểu thức điều kiện lặp&gt; trước, nếu <b>đúng</b> mới thực hiện &lt;khối lệnh xử lý lặp&gt;, nếu <b>sai</b> thì thoát khỏi vòng lặp.</li> <li>- Có thể <u>không thực hiện</u> &lt;khối lệnh xử lý lặp&gt; lần nào nếu &lt;biểu thức điều kiện lặp&gt; <b>sai</b> ngay từ đầu.</li> </ul>	<ul style="list-style-type: none"> <li>- Thực hiện &lt;khối lệnh xử lý lặp&gt; lần đầu, sau đó mới kiểm tra &lt;biểu thức điều kiện lặp&gt;, nếu <b>đúng</b> thì tiếp tục thực hiện &lt;khối lệnh xử lý lặp&gt;, nếu <b>sai</b> thì thoát khỏi vòng lặp.</li> <li>- <u>Thực hiện ít nhất 1 lần</u> &lt;khối lệnh xử lý lặp&gt; dù &lt;biểu thức điều kiện lặp&gt; <b>sai</b> ngay từ đầu.</li> </ul>

#### (4) CÁC BIẾN THỂ CỦA CÁC CÂU LỆNH LẶP

- Lệnh **for**:

a) Vừa khai báo vừa gán giá trị cho biến khởi động:

```
int S=0, n;
//Nhập liệu n>0
for(int i=1; i<=n; i++)
    S+=i;
printf("S = %d", S);
```

b) Di chuyển **1** lên trên vòng lặp

```
int S=0, n, i=1;
//Nhập liệu n>0
for(; i<=n; i++)
    S+=i;
printf("S = %d", S);
```

c) Di chuyển **4** vào trong vòng lặp sau **3**

```
int S=0, n, i=1;
//Nhập liệu n>0
for(; i<=n; )
{
    S+=i;
    i++;
}
printf("S = %d", S);
```

d) Di chuyển **(2)** vào trong vòng lặp trước **(3)** => trở thành lệnh **while**

```
int S=0, n, i=1;
//Nhập liệu n>0
for(;;)
{
    if(i>n)break;
    S+=i;
    i++;
}
printf("S = %d", S);
```

e) Di chuyển **(2)** vào trong vòng lặp sau **(4)** => trở thành lệnh **do ... while ...**

```
int S=0, n, i=1;
//Nhập liệu n>0
for(;;)
{
    S+=i;
    i++;
    if(i>n)break;
}
printf("S = %d", S);
```

- Lệnh while:**

Di chuyển **(2)** vào trong vòng lặp trước **(3)**

```
int S=0, n, i=1;
//Nhập liệu n>0
while(1) /*nếu while(0): không thực hiện khối lệnh xử lý trong
        while lần nào vì điều kiện đã sai (=0) ngay từ đầu.*/
{
    if(i>n)break;
    S+=i;
    i++;
}
printf("S = %d", S);
```

- Lệnh do ...while...:**

Di chuyển **(4)** vào trong vòng lặp sau **(3)**

```
int S=0, n, i=1;
//Nhập liệu n>0
do
{
    S+=i;
    i++;
    if(i>n)break;
}while(1); /*nếu while(0): thực hiện khối lệnh xử lý trong
        while lần đầu, sau đó thoát khỏi vòng lặp vì
        điều kiện lặp sai (=0).*/
printf("S = %d", S);
```

**Ví dụ 3:** làm bài tập có hướng dẫn số 1 chương 4 (sử dụng cả 3 câu lệnh lặp).

**Ví dụ 4:** Nhập n từ 1=>12. In giá trị tháng tương ứng ra màn hình (sử dụng cả 3 câu lệnh lặp để kiểm tra miền giá trị hợp lệ của n).

**Ví dụ 5:** Với CT mô phỏng trò chơi “Oẳn tù tì”, sau khi người chơi đã chơi 1 lần, hãy bổ sung lệnh hỏi người chơi có muốn tiếp tục chơi nữa không (C/K). Nếu người chơi chọn ‘C’ hoặc ‘c’ thì tiếp tục trò chơi. Nếu người chơi chọn ‘K’ hoặc ‘k’ thì thoát CT.

### Bài tập làm thêm:

1. Nhập n  $\in [2,9]$ . Viết CT in bảng cửu chương thứ n như sau:

```
Nhap n tu 2 => 9: -2
Nhap n tu 2 => 9: 10
Nhap n tu 2 => 9: 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

```
...
int n;
do
{
    printf("Nhap n tu 2 => 9: ");
    scanf("%d", &n);
}while(n<2||n>9);
for(int i=1;i<=10;i++)
    printf("%d x %2d = %2d\n",n,i,n*i);
...
```

2. Nhập m, n  $\in [2,9]$ . Viết CT in bảng cửu chương từ m đến n như sau:

```
Nhap m, n tu 2 => 9, m<=n, n-m<=4: 0 7
Nhap m, n tu 2 => 9, m<=n, n-m<=4: 2 9
Nhap m, n tu 2 => 9, m<=n, n-m<=4: 7 3
Nhap m, n tu 2 => 9, m<=n, n-m<=4: 3 6
3 x 1 = 3    4 x 1 = 4    5 x 1 = 5    6 x 1 = 6
3 x 2 = 6    4 x 2 = 8    5 x 2 = 10   6 x 2 = 12
3 x 3 = 9    4 x 3 = 12   5 x 3 = 15   6 x 3 = 18
3 x 4 = 12   4 x 4 = 16   5 x 4 = 20   6 x 4 = 24
3 x 5 = 15   4 x 5 = 20   5 x 5 = 25   6 x 5 = 30
3 x 6 = 18   4 x 6 = 24   5 x 6 = 30   6 x 6 = 36
3 x 7 = 21   4 x 7 = 28   5 x 7 = 35   6 x 7 = 42
3 x 8 = 24   4 x 8 = 32   5 x 8 = 40   6 x 8 = 48
3 x 9 = 27   4 x 9 = 36   5 x 9 = 45   6 x 9 = 54
3 x 10 = 30  4 x 10 = 40  5 x 10 = 50  6 x 10 = 60
```

```
...
int m,n;
do
{
    printf("Nhap m, n tu 2 => 9, m<=n, n-m<=4: ");
    scanf("%d%d", &m, &n);
}while(m<2||m>9||n<2||n>9||m>n||n-m>4);
```

```

for(int d=1;d<=10;d++)
{
    for(int c=m;c<=n;c++)
        printf("%d x %2d = %2d\t",c,d,c*d);
    printf("\n");
}
...

```

### 3. Viết CT in bảng cửu chương tổng hợp như sau:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

```

...
for(int d=1;d<=10;d++)
{
    for(int c=1;c<=10;c++)
        printf("%3d\t",c*d);
    printf("\n\n");
}
...

```