

Bài 8: Cấu trúc lệnh lặp for...

1. Cú pháp

```
for (<biểu thức 1>;< biểu thức 2>;< biểu thức 3>) {  
    <câu lệnh>;  
}
```

Trong đó:

<biểu thức 1> thường dùng để khởi gán giá trị cho biến đếm

<biểu thức 2> thường dùng để kiểm tra điều kiện lặp

<biểu thức 3> thường dùng để tăng biến đếm cho vòng lặp

2. Thực hiện

B1: Tính giá trị <biểu thức 1>.

B2: Tính giá trị <biểu thức 2>.

B3: Nếu giá trị <biểu thức 2> khác 0 (true) thì thực hiện <câu lệnh> sau đó tính giá trị <biểu thức 3> rồi quay lại B2. Còn nếu <biểu thức 2> =0 (false) thì thoát khỏi vòng lặp.

Chú ý:

+ <biểu thức 1> chỉ được tính 1 lần còn <biểu thức 2> và <biểu thức 3> được tính lại sau mỗi lần lặp.

+ <biểu thức 1>;< biểu thức 2>;< biểu thức 3> có thể vắng mặt tuy nhiên dấu (;) phải đặt đúng vị trí. Nếu < biểu thức 2> vắng mặt thì có nghĩa là điều kiện luôn đúng (vòng for không thoát).

3. Một số ví dụ

VD 1: Viết chương trình tính tổng bình phương N số nguyên dương đầu tiên (dùng cấu trúc lặp for).

```
#include <stdio.h>  
int n,s=0;  
int main(){  
    printf("N = ");  
    scanf("%d", &n);  
    s=0;
```

```

for (int i=1;i<=n;i++) {
    s+=i*i;
}
printf("Tong = %d", s);
return 0;
}

```

VD 2: viết chương trình tính N!

```

#include <stdio.h>
int n,gt;
int main() {
    printf("N = ");
    scanf("%d", &n);
    gt=1;
    for (int i=1;i<=n;i++) {
        gt=gt*i;
    }
    printf("Giai thua cua %d la %d", n, gt);
    return 0;
}

```

VD 3: Viết chương trình in bảng cửu chương từ 2->9 ra màn hình.

```

#include <stdio.h>
int main() {
    for (int i=2;i<=9;i++) {
        printf("Ban cuu chuong %d:\n", i);
        for (int j=1;j<=10;j++) {
            printf("%d x %d = %d\n", j, i, i*j );
        }
    }
    return 0;
}

```