



# HUST

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.





ĐẠI HỌC  
BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

# C PROGRAMMING LANGUAGE

## ĐIỀU KHIỂN LẶP (PHẦN 1)

ONE LOVE. ONE FUTURE.

# VÒNG LẶP

- Được sử dụng để thực hiện lặp nhiều lần cùng một lệnh hay khối lệnh
- C hỗ trợ tạo vòng theo nhiều cách thức mềm dẻo để quyết định số lần lặp và khi nào thì kết thúc vòng lặp
- Các loại vòng lặp hỗ trợ trong ngôn ngữ C là
  - for
  - while
  - do ... while

# LỆNH WHILE

**while ( <điều kiện> )**  
**<câu lệnh>**

- Thực hiện lặp một **lệnh** hay một **khối lệnh** theo một điều kiện
- Kiểm tra điều kiện ngay khi bắt đầu vào vòng lặp
- Kết thúc vòng lặp khi điều kiện trở nên **sai** (có giá trị bằng 0)

# VÍ DỤ

- Đọc một tập số nguyên và in tổng của chúng

```
tổng = 0
đếm = 0

input sốpt

while (đếm < sốpt) do
{
    input sốsau
    cộng số sau vào tổng
    thêm 1 vào đếm
}
```

```
#include <stdio.h>
int main()
{
    int tong=0, dem=0, sopt, sosau;

    printf("So phan tu trong day so:");
    scanf("%d", &sopt);

    while (dem < sopt)
    {
        scanf("%d", &sosau);
        tong += sosau;
        dem++;
    }

    printf("Tong la %d\n",tong);
    return 0;
}
```

Không có  
do ở đây

# VÍ DỤ (TIẾP)

```
#include <stdio.h>
int main()
{
    int tong=0, dem=0, sopt, sosau;

    printf("So phan tu trong day so:");
    scanf("%d", &sopt);

    while (dem < sopt)
    {
        scanf("%d", &sosau);
        tong += sosau;
        dem++;
    }

    printf("Tong la %d\n",tong);
    return 0;
}
```

sopt	sosau	tong	dem
?	?	0	0
3			
	4	4	1
	-1	3	2
	2	5	3

# LỖI THƯỜNG GẶP

```
while (dem < sopt)
    scanf("%d", &sosau);
    tong += sosau;
    dem++;
```

Chỉ có lệnh scanf là  
được lặp nhiều lần

```
while (dem < sopt);
{
    scanf("%d", &sosau);
    tong += sosau;
    dem++;
}
```

Lệnh lặp chỉ là lệnh  
rỗng (lệnh chỉ có  
dấu ;)

```
while (dem < sopt)
{
    scanf("%d", &sosau);
    tong += sosau;
    dem++;
    printf("Tong la %d\n", tong);
}
```

Lệnh in bị lặp lại  
nhiều lần



# KẾT THÚC LẬP VỚI EOF

- Trong ví dụ tính tổng dãy số trước, để xác định sự kết thúc của dãy số ta phải xác đếm số phần tử trong dãy trước khi bắt đầu nhập dãy.
- Có một cách xác định kết thúc của một dãy số được nhập là thông qua dấu hiệu EOF trả về bởi hàm scanf.
- Trong quá trình nhập dữ liệu, nếu người sử dụng nhấn phím tắt Ctrl+D trong Unix (hoặc Ctrl+Z trong DOS) thì hàm scanf sẽ trả về EOF.

# VÍ DỤ

- Đọc một tập số nguyên và in tổng của chúng (ver 2)

```
tổng = 0

while (chưa kết thúc dãy) do
{
    input sốsau
    cộng sốsau vào tổng
}

output tổng
```

```
#include <stdio.h>

int main()
{
    int aNum, sum = 0;

    while (scanf("%d", &aNum) != EOF)
    {
        sum += aNum;
    }

    printf("Sum is %d\n", sum);
    return 0;
}
```

# LỆNH FOR

**for** ( <khởi tạo> ; <điều kiện> ; <cập nhật> )  
    <câu lệnh>

- Là một dạng của vòng lặp **while** với các thành phần *khởi tạo* và *cập nhật* được chỉ ra trong câu lệnh
- Các thành phần trong câu lệnh **for** là **tuỳ chọn**. Khi điều kiện lặp không được chỉ rõ thì nó được lấy giá trị ngầm định là **đúng**.
- Cập nhật luôn thực hiện sau câu lệnh của vòng lặp

- Đọc một tập số nguyên và in tổng của chúng

```
tổng = 0
đếm = 0

input sốpt

while (đếm < sốpt) do
{
    input sốsau
    cộng sốsau vào tổng
    thêm 1 vào đếm
}
```

```
#include <stdio.h>
int main()
{
    int tong=0, dem, sopt, sosau;

    printf("So phan tu trong day so:");
    scanf("%d", &sopt);

    for (dem=0; dem<sopt; dem++)
    {
        scanf("%d", &sosau);
        tong += sosau;
    }

    printf("Tong la %d\n",tong);
    return 0;
}
```

# SO SÁNH WHILE VÀ FOR

```
#include <stdio.h>
int main()
{
    int tong=0, dem;
    int sopt, sosau;

    printf("So phan tu: ");
    scanf("%d", &sopt);
    dem = 0;
    while (dem < sopt)
    {
        scanf("%d", &sosau);
        tong += sosau;
        dem++;
    }
    printf("Tong la %d\n",tong);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    int tong=0, dem;
    int sopt, sosau;

    printf("So phan tu: ");
    scanf("%d", &sopt);

    for (dem=0; dem<sopt; dem++)
    {
        scanf("%d", &sosau);
        tong += sosau;
    }

    printf("Tong la %d\n",tong);
    return 0;
}
```

# LỖI THƯỜNG GẶP

```
for (dem=0; dem<sopt; )  
{  
    scanf("%d", &sosau);  
    tong += sosau;  
}
```

Biến đếm không được cập nhật sau mỗi lần lặp

```
for (dem=0; dem<sopt; dem++);  
{  
    scanf("%d", &sosau);  
    tong += sosau;  
}
```

Dấu ; không có ở đây

```
for (dem=0, dem<sopt, dem++)  
{  
    scanf("%d", &sosau);  
    tong += sosau;  
}
```

Dấu phẩy không phải là dấu phân cách các thành phần

# DẤU PHẨY

- Trong vòng for dấu phẩy (,) được dùng để phân cách nhiều lệnh trong cùng một nhóm

- Ví dụ:

```
for (i=0, j=100; i<=j; i++, j--)  
    printf("(%d, %d\n)", i, j);
```

Output:

(0, 100)

(1, 99)

...

(49, 51)

(50, 50)

1. Viết chương trình in các số có 2 chữ số có tổng là 10, ví dụ 19, 28,...
2. Viết chương trình in 100 số đầu tiên của dãy số sau: 1 2 3 5 8 13 21...
3. Viết chương trình nhận vào một số nguyên  $n \leq 9$  và in ra một tam giác có dạng như sau với  $n = 5$ .

1

12

123

1234

12345



# LỜI GIẢI (BÀI 1)

```
for (x=1; x<=9; x++)  
{  
    printf("%d%d\n", x, 10-x);  
}
```

# LỜI GIẢI (BÀI 2)

```
first = 1; second = 2;
for (count=1; count<=100; count++)
{
    printf("%5d", first);
    tmp = first + second;
    first = second;
    second = tmp;
}
```

# LỜI GIẢI (BÀI 3)

```
for (i=1; i<=n; i++)  
{  
    for (j=1; j<=i; j++)  
        printf("%d", j);  
    printf("\n");  
}
```

A graphic on the left side of the slide. It features a dark blue background with a large, stylized circular shape composed of many small red dots. The dots are arranged in a way that creates a sense of depth and movement, with some dots appearing larger and more concentrated than others. In the center of this graphic, the word "HUST" is written in a bold, white, sans-serif font.

**HUST**

**THANK YOU !**