



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

C PROGRAMMING INTRODUCTION

TUẦN 7: VÒNG LẶP

Nội dung

- Vòng lặp
 - Ôn tập
 - Cấu trúc lặp
 - Các chú ý
 - Thực hành

Cấu trúc lặp **for**

- Định dạng cấu trúc lặp **for**

for (*khởi tạo* ; điều kiện tiếp tục ; *cập nhật*)
câu lệnh

Ví dụ:

```
for( int counter = 1; counter <= 10; counter++ )  
    printf( "%d\n", counter );
```

- In các số nguyên từ 1 đến 10

Không có
dấu ; ở
cuối

Cấu trúc lặp **for** (II)

- Vòng lặp **For** có thể viết thành vòng lặp **while**:

khởi tạo;

```
while (điều kiện tiếp tục){  
    câu lệnh  
    cập nhật;  
}
```

- Khởi tạo và cập nhật

- Các câu lệnh khởi tạo cách nhau bởi dấu ,

```
for (int i = 0, j = 0; j + i <= 10; j++, i++)  
    printf( "%d\n", j + i );
```

Cấu trúc For: Các chú ý

- Biểu thức số học:
 - khởi tạo, điều kiện tiếp tục và cập nhật có thể chứa các biểu thức số học. If $\mathbf{x} = 2$ and $\mathbf{y} = 10$
 $\mathbf{for} \ (\ j = \mathbf{x}; \ j \leq 4 * \mathbf{x} * \mathbf{y}; \ j += \mathbf{y} / \mathbf{x} \)$
tương đương với
 $\mathbf{for} \ (\ j = 2; \ j \leq 80; \ j += 5 \)$
- "cập nhật" có thể theo chiều âm (giảm)
- Nếu điều kiện tiếp tục có giá trị khởi tạo **false**
 - Thân của cấu trúc **for** sẽ không được thực hiện
 - Các câu lệnh sau cấu trúc **for** được thực hiện

Cấu trúc For: Các chú ý (II)

- Biến điều khiển
 - Thường được in hoặc sử dụng bên trong thân vòng **for** nhưng không bắt buộc
- **For** có lược đồ tương tự **while**

Ví dụ

• Ví dụ For

```
for (i=1;i<=100;i++) {  
    x += i;  
    if ((x % i) == 0) { i--; }  
}
```

```
for (i=0, j=strlen(s)-1; i<j; i++,j--)  
    { c = s[i], s[i] = s[j], s[j] = c; }
```

```
char c;  
int count;  
for (count=0; (c=getchar()) != '\n'); count++)  
    { }  
printf("Number of characters is %d\n", count);
```

Bài tập 7.1

- Viết chương trình in ra mười số nguyên và bình phương của chúng

1 1

2 4

3 9

...

10 100

Lời giải

```
#include <stdio.h>
```

```
int main()
```

```
{  
    int i;
```

```
    for(i = 1; i <= 10; i = i + 1)  
        printf("%d %d\n", i, i * i);
```

```
    return 0;
```

```
}
```

Bài tập 7.2

- Viết chương trình in ra tam giác như sau:

*

**

Lời giải

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i, j;
```

```
    for(i = 1; i <= 10; i = i + 1)
```

```
    {
```

```
        for(j = 1; j <= i; j = j + 1)
```

```
            printf("*");
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

Bài tập 7.3

- Viết chương trình in ra số lượng các số lớn hơn 27 trong các số từ 1 đến 100

Lời giải

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int i;
```

```
int count = 0;
```

```
for(i = 1; i <= 100; i = i + 1)
```

```
{
```

```
if(i > 27)
```

```
    count = count + 1;
```

```
}
```

```
printf("%d numbers were greater than 27\n", count);
```

```
return 0;
```

```
}
```

Bài tập 7.4

- Viết chương trình in ra các số nguyên tố ≤ 100
- Sử dụng thư viện `math.h` để dùng các hàm toán học: `sqrt`,...

Lời giải

```
#include <stdio.h>
#include <math.h>
```

```
main()
{
    int i, j;
```

```
    printf("%d\n", 2);
```

```
    for(i = 3; i <= 100; i = i + 1)
    {
        for(j = 2; j < i; j = j + 1)
        {
            if(i % j == 0)
                break;
```

Lời giải

```
if(j > sqrt(i))
{
    printf("%d\n", i);
    break;
}
}
}
return 0;
}
```


Bài tập 7.5

- Sửa bài tập 7.4 bằng cách bỏ đi các số chẵn để không phải gọi hàm sqrt nhiều lần.

Lời giải

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main()
```

```
{
```

```
int i, j;
```

```
double sqrti;
```

```
printf("%d\n", 2);
```

```
for(i = 3; i <= 100; i = i + 2)
```

```
{
```

```
    sqrti = sqrt(i);
```

Lời giải

```
for(j = 2; j < i; j = j + 1)
{
if(i % j == 0)
    break;                /* not prime */
if(j > sqrt(i))
    {                    /* prime */
    printf("%d\n", i);
    break;
    }
}

return 0;
}
```

Bài tập 7.6

- Gõ thử chương trình và biên dịch

```
/* Counting down to blast-off */
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int time, start;
```

```
    printf("Enter starting time (an integer) in seconds> ");
```

```
    scanf("%d", &start);
```

```
    printf("\nBegin countdown\n");
```

```
    for (time = start; time > 0; time = time - 1)
```

```
    {
```

```
        printf("T - %d\n", time);
```

```
    }
```

```
    printf("Blast-off!\n");
```

```
    return (0);
```

```
}
```

Bài tập 7.7

- Viết chương trình in ra bảng chuyển đổi nhiệt độ Celsius sang Fahrenheit.
- Chú ý điều kiện tiếp tục vòng lặp và sử dụng `#define` để định nghĩa hằng số
- $\text{fahrenheit} = 1.8 * \text{celsius} + 32.0;$

Lời giải

```
#include <stdio.h>
```

```
/* Constant macros */
```

```
#define CBEGIN 10
```

```
#define CLIMIT -5
```

```
#define CSTEP 5
```

```
int
```

```
main(void)
```

```
{
```

```
    /* Variable declarations */
```

```
    int    celsius;
```

```
    double fahrenheit;
```

```
    /* Print the table heading */
```

```
    printf(" Celsius Fahrenheit\n");
```

Lời giải

```
/* Print the table */  
    for (celsius = CBEGIN;  
        celsius >= CLIMIT;  
        celsius = celsius + CSTEP) {  
        fahrenheit = 1.8 * celsius + 32.0;  
        printf(" %3d  %7.2f\n", celsius, fahrenheit);  
    }  
  
    return (0);  
}
```

Bài tập 7.8

- Gõ chương trình và biên dịch để hiểu cơ chế vòng lặp lồng nhau (vòng lặp ở bên trong vòng lặp)

exercise7_8.c

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int i, j;                /* loop control variables */
```

```
    printf("      I  J\n");    /* prints column labels */
```

```
    for (i = 1; i < 4; i = i + 1)    /* heading of outer for loop */
```

```
    {
```

```
        printf("Outer %6d\n", i);
```

```
        for (j = 0; j < i; j = j + 1) /* heading of inner loop */
```

```
        {
```

```
            printf("  Inner%9d\n", j);
```

```
        }
```

```
            /* end of inner loop */
```

```
    }    /* end of outer loop */
```

```
    return (0);
```

```
}
```

Bài tập 7.9

- Viết chương trình sử dụng cấu trúc for để tính n giai thừa
- VD:

Results

```
Enter n: 4  
4! = 24
```

Results

```
Enter n: 0  
0! = 1
```

Lời giải

```
# include <stdio.h>
int main ()
{
    int i, n, f;
    printf (" Enter n: ");
    scanf ("%d", &n);
    f = 1; /* 0! */
    for (i = 1; i <= n; ++i) {
        f *= i; /* Now , f = i! */
    }
    printf ("%d! = %d\n", n, f);
    return 0;
}
```

Bài tập 7.10

- Trong số học, số hoàn hảo là số nguyên dương có giá trị bằng tổng các ước số nguyên của nó (không kể chính nó). E.g: $6=1+2+3$
- Viết chương trình in ra các số hoàn hảo có giá trị $\leq N$ được người dùng nhập vào

Lời giải

```
#include <stdio.h>

void main()
{
    int n, i, j, tong;

    printf("\nEnter N= : ");
    scanf("%d", &n);
    for (i=2; i<=n; i++)
    {
        tong = 1;
        for (j=2; j<=i/2; j++)
            if (i%j == 0)
                tong += j;
        if (tong == i)
            printf("\n%d", i);
    }
}
```



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

