

Ngôn ngữ lập trình C

B7: Vòng lặp



PHENIKAA
UNIVERSITY

Khoa Công nghệ thông tin

CHỦ ĐỀ

- **Vòng lặp**

- Cấu trúc vòng lặp `FOR` biết trước số lần lặp
- Bài tập thực hành

Cấu trúc vòng lặp FOR

- **Gợi mở:**

- Viết chương trình in ra các số nguyên từ **1 đến 5**, mỗi số trên 1 dòng.
- Chương trình đơn giản? Sử dụng những kỹ năng được học từ buổi đầu tiên đến nay thì chỉ cần dùng 5 lệnh printf liên tiếp.
- Thay đổi yêu cầu bài toán: vẫn dùng phương pháp ở trên áp dụng cho việc in ra 10 số từ **1 đến 10**? Mức độ khó?, mức độ phức tạp?, có làm được không?
- Thay đổi yêu cầu bài toán: vẫn dùng phương pháp ở trên áp dụng cho việc in ra 100 số từ **1 đến 100**? Mức độ khó?, mức độ phức tạp?, có làm được không?
- Thay đổi yêu cầu bài toán: vẫn dùng phương pháp ở trên áp dụng cho việc in ra 1000 số từ **1 đến 1000**? Mức độ khó?, mức độ phức tạp?, có làm được không?
- Thay đổi yêu cầu bài toán: vẫn dùng phương pháp ở trên áp dụng cho việc in ra 10000 số từ **1 đến 10000**? Mức độ khó?, mức độ phức tạp?, có làm được không?

Cấu trúc vòng lặp FOR

- **Mục đích:** dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp xác định.

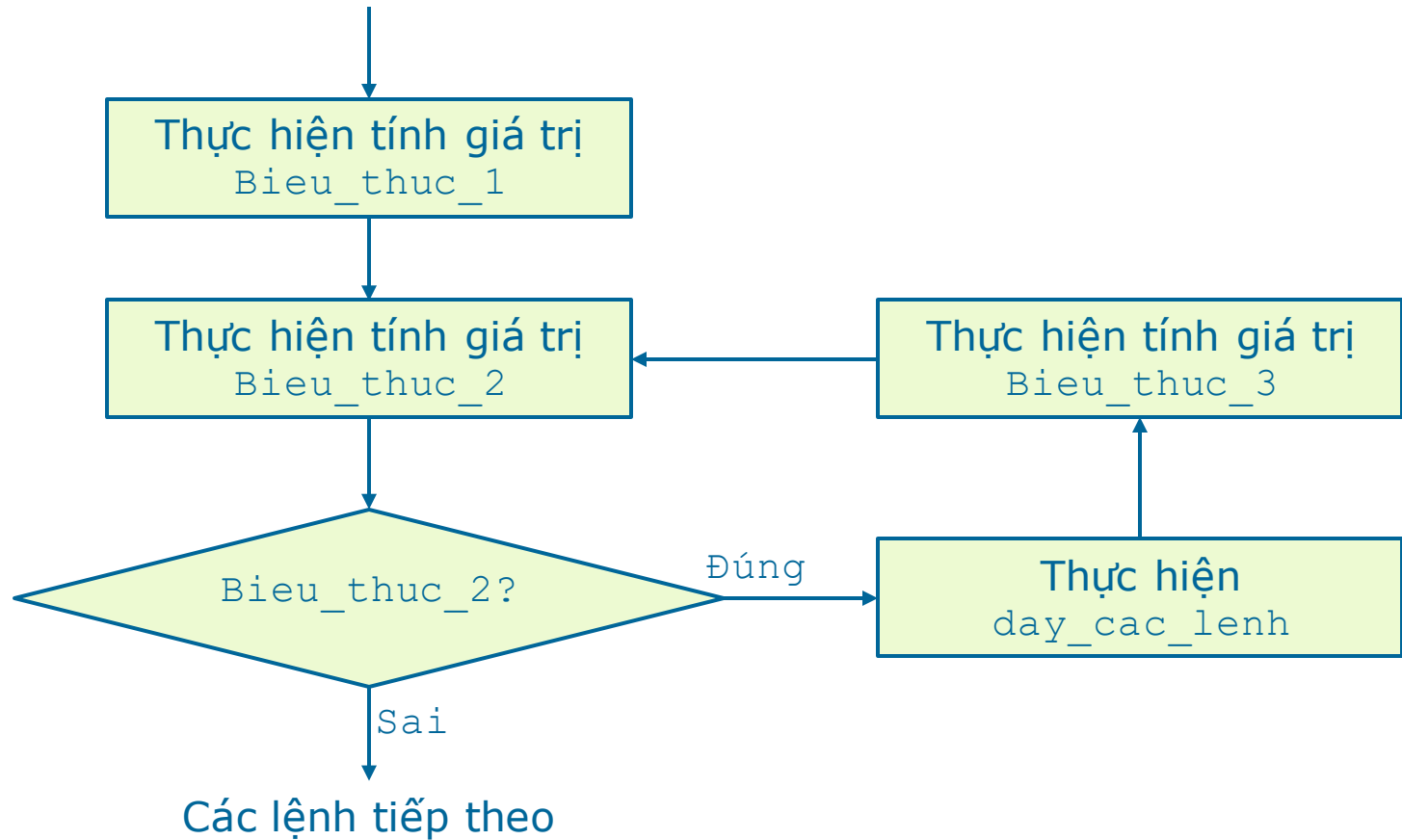
- **Cú pháp:**

```
for(bieu_thuc_1;bieu_thuc_2;bieu_thuc_3)
{
    day_cac_lenh;
}
```

- **Trong đó:**

- bieu_thuc_1: Khởi tạo giá trị ban đầu cho vòng lặp;
- bieu_thuc_2: Điều kiện tiếp tục vòng lặp;
- bieu_thuc_3: Thực hiện bước tăng của vòng lặp;
- Chú ý các biểu thức 1, 2, 3 có thể có hoặc không.

Cấu trúc vòng lặp FOR



Lưu ý đối với cấu trúc FOR

- Với `bieu_thuc_1` và `bieu_thuc_3` có thể là một danh sách lệnh ngăn cách nhau bởi dấu phẩy (,).
- Ví dụ:

```
for (int i = 0, j = 0; j + i <= 10; j++, i++)  
    printf( "%d\n", j + i );
```
- Nếu `bieu_thuc_1` trong quá trình khởi tạo sai thì phần `danh_sach_lenh` sẽ không được thực hiện một lần nào

Lưu ý đối với cấu trúc FOR

- Cấu trúc `for` lồng nhau:
- Ví dụ 7.1:

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

- Lệnh `break` dùng để thoát mọi vòng lặp. Cải tiến ví dụ trên dùng lệnh `break` để thoát vòng lặp nếu $i, j > 5$.

Ví dụ 7.1

```
#include<stdio.h>
void main(){
    int i, j, N;
    printf("Nhap vao so nguyen duong N = ");
    scanf("%d",&N);
    for(i=1;i<=N;i++)
    {
        if(i>5) break;
        for(j=1;j<=N;j++)
        {
            if(j>5) break;
            printf("(%d,%d)\t",i,j);
        }
        printf("\n");
    }
}
```


Ví dụ 7.2

- In ra màn hình các số nguyên từ 1 đến 100

```
#include<stdio.h>
void main(){
    int i;
    for(i=1;i<=100;i++)
        printf("%5d",i);
    printf("\n");
}
```

Ví dụ 7.3

- In ra màn hình 2 hàng số nguyên: hàng trên là các số nguyên chẵn và hàng dưới là các số nguyên lẻ.

```
#include<stdio.h>
void main(){
    int i;
    printf("Cac so chan:\n");
    for(i=2;i<=30;i+=2)
        printf("%5d",i);
    printf("\nCac so le:\n");
    for(i=1;i<=30;i+=2)
        printf("%5d",i);
    printf("\n");
}
```

Ví dụ 7.4

- Nhập vào một số nguyên dương n , yêu cầu kiểm tra điều kiện $n > 0$. Nếu $n < 1$ thì thoát chương trình với thông báo n không thỏa mãn điều kiện ngược lại thì in ra màn hình theo dạng ma trận dạng như sau (ví dụ $n = 3$):

1	1	1
2	2	2
3	3	3

Ví dụ 7.4

- Đáp án

```
#include<stdio.h>
void main(){
    int i, j, n;
    printf("Nhap vao so nguyen duong n = ");
    scanf("%d",&n);
    if(n < 1) printf("n nhap vao khong hop le!\n");
    else
    {
        printf("Ma tran %dx%d:\n",n,n);
        for(i=1;i<=n;i++)
        {
            for(j=1;j<=n;j++)
                printf("%5d",i);
            printf("\n");
        }
    }
}
```

Ví dụ 7.5

- **Giải bài toán vui:**

Vừa gà vừa chó

Bó lại cho tròn

Ba mươi sáu (36) con

Một trăm (100) chân chẵn

Hỏi mấy gà, mấy chó?

```
#include<stdio.h>
#define TONGGACHO 36
void main(){
    int i, j, ga, cho;
    printf("Vua ga vua cho\n");
    printf("Bo lai cho tron\n");
    printf("Ba muoi sau (36) con\n");
    printf("Mot tram (100) chan chan\n");
    printf("Hoi may ga?, may cho?\n");
    printf("Dap an la:\n");
    for(i=1;i<=TONGGACHO;i++)
        for(j=1;j<=TONGGACHO;j++)
            if((i+j == TONGGACHO) && (i*2+j*4 == 100))
            {
                ga = i;
                cho = j;
                break;
            }
    printf("So Ga = %d; so Cho = %d\n",ga, cho);
}
```

Bài tập 7.1

- Liệt kê các số tự nhiên nhỏ hơn 1000 có đặc tính sau:
 - tổng các chữ số hàng đơn vị và trăm bằng chữ số hàng chục
 - Là số chia hết cho 3
 - Cứ 10 số lại xuống dòng

Bài tập 7.2

- **Tiền lãi gửi tiết kiệm:** Ngân hàng ACB ra chương trình khuyến mãi gửi tiền có kỳ hạn một năm, mỗi tháng lãi suất f (VD 1.2)%. f – thực – nhập từ bàn phím. Tiền lãi hàng tháng được cộng dồn vào tiền gửi. Hãy viết chương trình yêu cầu người dùng nhập tên và số tiền gửi, lãi suất sau đó in ra bảng tiền lãi và tiền trong sổ sau từng tháng.

- **Ví Dụ:**

Tài khoản khách hàng: Nguyễn Thị Tâm

Tháng	Tiền đầu kỳ	Tiền lãi	Số dư
1	1000000	12000	1012000
2	1012000
...			
12			

Tổng lãi:

Bài tập 7.3

- Viết chương trình in ra màn hình hình dạng như sau:

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

Cải tiến để nhập số dòng cần in ra từ bàn phím.

Bài tập 7.4

- In ra màn hình:
 - a) tam giác vuông ngược chiều cao n (nhập từ bàn phím) như sau:

* * * *

* * *

* *

*

- b) tam giác cân chiều cao n (nhập từ bàn phím) như sau:

*

* * *

* * * * *

Bài tập

5. Viết chương trình in ra bảng chữ cái.
6. Tính $n!$ Với n nguyên nhập từ bàn phím và $n < 8$.
7. Viết chương trình tính tổng các số nguyên từ 1 đến n , với n nhập từ bàn phím.
8. Viết chương trình tính tổng các số nguyên chẵn hoặc lẻ từ 1 đến n , với n nhập từ bàn phím. Viết giao diện cho phép người dùng lựa chọn muốn tính tổng số chẵn hoặc số lẻ.

Bài tập

9. Lập trình tính tổng:

$$S=1+\frac{1}{2}+\frac{1}{3}+\dots+\frac{1}{n}$$

Với n là số nguyên dương nhập từ bàn phím.

10. Viết chương trình kiểm tra một số nguyên nhập từ bàn phím có phải là số nguyên tố hay không?

Bài tập 7.11

- Viết chương trình thực hiện các công việc sau:
 - Nhập số `epsilon < 1` từ bàn phím
 - Tính số `e` theo công thức:
$$e = 1 + 1/(1!) + 1/(2!) + \dots + 1/(n!)$$
 - Quá trình tính dừng khi $1/(n!) < \text{epsilon}$
 - Đưa kết quả ra màn hình.

Bài tập 7.12

- Tính số π theo công thức:

$$\pi / 4 = 1 - 1/3 + 1/5 - 1/7 + \dots + (-1)^i * 1 / (2*i+1)$$

- với độ chính xác `epsilon` nhập từ bàn phím.
Khi đó ta sẽ có số π với sai số `epsilon` khi i đủ lớn để:

$$1 / (2*i+1) < \text{epsilon}$$

Bài tập 7.13

- Giải câu đố “Trăm cỏ trăm trâu, trâu đứng ăn năm – trâu nằm ăn ba, lụ khụ trâu già ba trâu 1 bó”
- Số trâu mỗi loại và chi tiết số bó cỏ mà mỗi loại đã ăn.
- Cố gắng tối ưu số vòng lặp – không vét cạn.