

Ngôn ngữ lập trình C

B8: Vòng lặp



PHENIKAA
UNIVERSITY

Khoa Công nghệ thông tin

CHỦ ĐỀ

- **Vòng lặp**

- Cấu trúc vòng lặp điều kiện không biết trước số lần lặp
- Câu lệnh `continue` và `break`
- Bài tập thực hành

Gợi mở

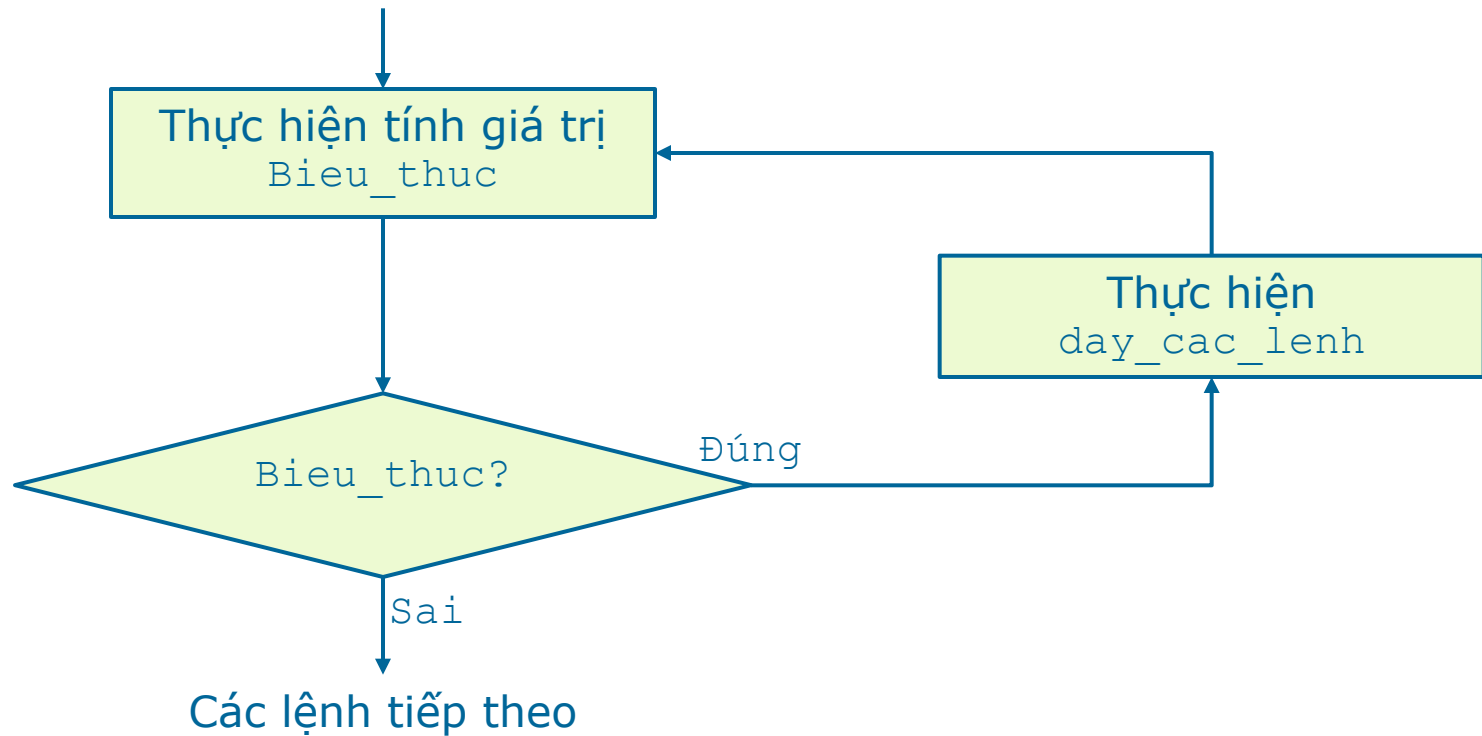
- Cấu trúc lặp `for`: biết trước là sẽ phải lặp đi lặp lại bao nhiêu lần.
- Bài toán ví dụ:
 - Giảng viên yêu cầu sinh viên làm 10 bài tập mỗi buổi học. **Cấu trúc lặp ???**
 - Giảng viên yêu cầu sinh viên thực hành đến khi thành thạo. **Cấu trúc lặp ???** :
 - Trường hợp làm vài bài thực hành đã thành thạo.
 - Trường hợp làm rất nhiều bài thực hành mới thành thạo.

Cấu trúc `while`

- Cú pháp:

```
while (bieu_thuc)
{
    day_cac_lenh;
}
```

Cấu trúc while



Cấu trúc `while`

- Mục đích:

- Dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp không xác định.
- Kiểm tra điều kiện vòng lặp (tức là giá trị của biểu thức) trước rồi mới thực hiện lệnh.
- Các lệnh sau `while` có thể không được thực hiện lần nào.

Cấu trúc `while`

- Ví dụ 8.1: Viết chương trình nhập vào các số nguyên cho đến khi nhập số -1 thì dừng lại.

```
2  #include<stdio.h>
3  int main()
4  {
5      int i;
6      printf("Nhap vao cac so, nhap -1 de ket thuc:");
7      scanf("%d",&i);
8      while(i!=-1)
9      {
10         scanf("%d",&i);
11     }
12     return 0;
13
14 }
```

Cấu trúc `while`

- **Ví dụ 8.2:** Nhập vào điểm của một sinh viên, nếu điểm đó không thuộc đoạn `[0, 10]` thì thông báo cho người dùng nhập lại.
- Cách làm:
 - Nếu dùng lệnh `if` thì chỉ kiểm tra được 1 lần;
 - Không dùng `for` được vì chưa biết trước số lần lặp.
 - Sử dụng vòng lặp `while`.

Cấu trúc while

- Đáp án ví dụ 8.2:

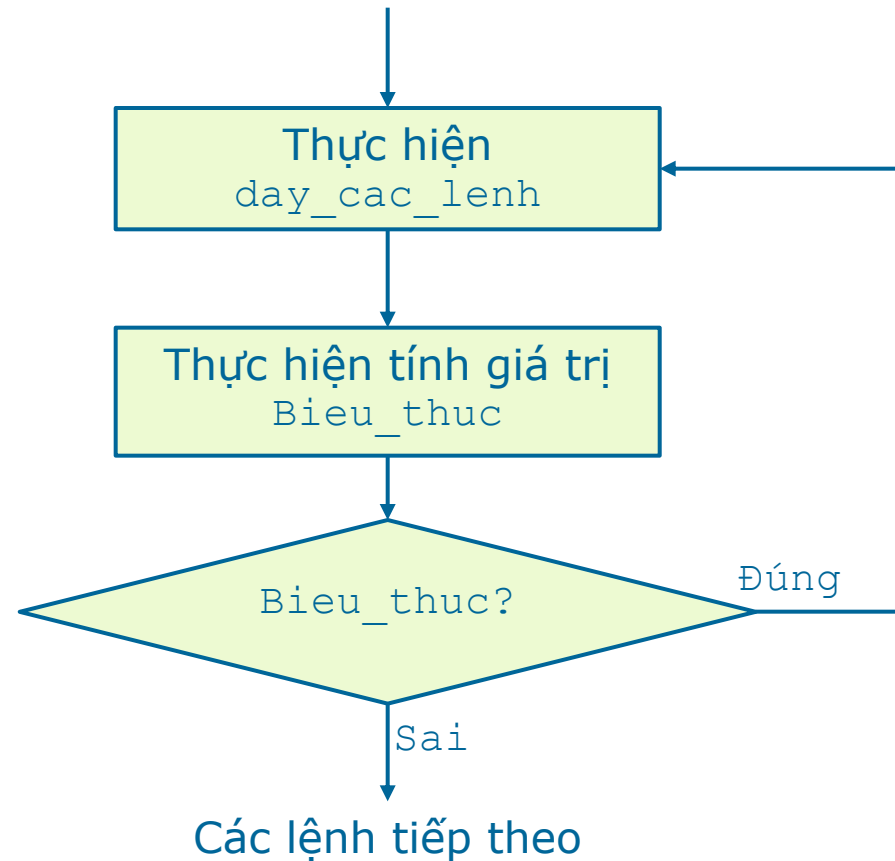
```
2  #include<stdio.h>
3  int main()
4  {
5      float d;
6      printf("Nhap diem cua sinh vien:");
7      scanf("%f", &d);
8      while(d<0 || d>10)
9      {
10         printf("Ban da nhap diem khong hop le:");
11         printf("Moi ban nhap lai diem sinh vien:");
12         scanf("%f",&d);
13
14     }
15     printf("Diem cua sinh vien vua nhap: %.2f\n",d);
16     return 0;
17 }
```

Cấu trúc do ... while

- Cú pháp:

```
do
{
    day_cac_lenh;
} while (bieu_thuc);
```

Cấu trúc do ... while



Cấu trúc do ... while

- Mục đích:

- Dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp không xác định.
- Thực hiện công việc nào đó xong rồi mới kiểm tra điều kiện vòng lặp (tức là giá trị của biểu thức).
- Trong cấu trúc do ... while, `day_cac_lenh` được thực hiện ít nhất một lần.

Cấu trúc do ... while

- Ví dụ 8.3

```
int i = 1, sum = 0;  
do {  
    sum += i;  
    i++;  
} while (i <= 50);  
printf("Tong tu 1 den 50 la %d\n", sum);
```



Kết quả?

Cấu trúc do ... while

- Ví dụ 8.4: in ra các số từ 1 đến N, với N nhập từ bàn phím.

```
int i = 1, N;  
printf("Nhap N: ");  
scanf("%d", &N);  
do {  
    printf(" %d ", i);  
} while (++i <= N);
```

Cấu trúc do ... while

- Ví dụ 8.5: Cải tiến ví dụ 8.1 và 8.2 từ cấu trúc while thành do ... while.

Các lệnh thay đổi cấu trúc lập trình

- Đối với các lệnh lặp: `while`, `do{...} while`, hoặc `for`, thay đổi việc thực hiện lệnh trong vòng lặp → C cung cấp 2 lệnh:
 - `continue;`
 - `break;`

Các lệnh thay đổi cấu trúc lập trình

- `continue`
 - Bỏ qua việc thực hiện các câu lệnh nằm sau lệnh `continue` trong thân vòng lặp.
 - Chuyển sang thực hiện một vòng lặp mới.
- `break`
 - Thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.

Ví dụ 8.6 continue và break

```
1  #include<stdio.h>
2  int main()
3  {
4      char c;
5      printf("Nhap vao cac ky tu (go S (STOP) de dung lai):");
6      while(1)
7      {
8          c = getchar();
9          if(c == 'S') break;
10         else if(c >='0' && c<='9') continue;
11         else putchar(c);
12     }
13     printf("END");
14     return 0;
15 }
```

Tạo menu tương tác sử dụng `while` hoặc `do ... while` kết hợp `switch ... case`

```
char ch; /* int c */
do {
    ch = getchar(); /* scanf(%d, &c) */
    switch (ch) {
        case: 'A'
            /* do some thing */ break;
        case: 'B'
            /* do some thing */ break;

        ...
        case: 'Q'
            Print Quit; break;
    }
} while (ch!='Q');
```

Kiểm soát nhập liệu

- ```
do {
 - printf("Nhap vao n:");
 - scanf("%d", &n);
 - if (n khong thoa man)
 printf ("Canh bao\n");
}
while (n khong thoa man)
```

# BÀI TẬP

1. Viết chương trình tính  $n!$  (kiểm soát  $0 < n < 8$ ) bằng 2 cấu trúc `while` và `do ...while`.
2. Lập chương trình thực hiện các công việc sau: Nhập số nguyên  $10 < n < 100$  bất kỳ. Kiểm tra sự hợp lệ của dữ liệu đầu vào. Nếu người dùng nhập sai yêu cầu nhập lại ngay. Nếu dữ liệu hợp lệ thì tính tổng các chữ số từ 1 đến  $n$ . Hiển thị kết quả ra màn hình.

# BÀI TẬP

## 3. 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

4. Viết chương trình dùng `while` hoặc `do . . . while` để tính số  $\pi$  theo công thức:

- Nhập số `epsilon < 1` từ bàn phím;
- Tính số  $\pi$  theo công thức:

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

- quá trình tính dừng khi:

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

- Đưa kết quả ra màn hình.

# BÀI TẬP

5. Viết chương trình tính  $\sin(x)$  với  $x$  là số thực nhập vào từ bàn phím theo công thức sau với độ chính xác 0.0001:

$$\begin{aligned} \sin(x) = & x - x^3/3! + x^5/5! + \dots \\ & + (-1)^n * x^{2*n+1} / (2*n+1)! \end{aligned}$$



# BÀI TẬP

6. Viết chương trình nhập vào 2 số nguyên dương từ bàn phím và đưa ra ước số chung lớn nhất của chúng.
7. Viết chương trình nhập vào 2 số nguyên từ bàn phím và đưa ra bội số chung nhỏ nhất của chúng.
8. Viết chương trình tính tích phân với độ chính xác  $10^{-4}$ :

$$\int_0^1 \frac{\sin(x^2)}{e^x} dx$$

# BÀI TẬP

9. Viết chương trình tạo ra một menu tương tác cho phép người dùng lựa chọn các phép tính số học: cộng, trừ, nhân, chia. Đối với phép chia thì phải kiểm soát chia 0. Sau khi tính xong thì đưa ra câu hỏi xem người dùng có tiếp tục tính tiếp hay không, lặp cho đến khi người dùng muốn thoát. Ví dụ:

```
CHUONG TRINH TINH
```

```
1- Phep cong
```

```
2- Phep tru
```

```
3- Phep nhan
```

```
4- Phep chia
```

```
An phim khac: Thoat
```

```
Moi ban lua chon chuong trinh: 1
```

```
Ban da chon chuong trinh tinh tong
```

```
Nhap so thu 1: 4
```

```
Nhap so thu 2: 21
```

```
Tong hai so la: 25
```

```
Ban co muon tiep tục chuong trinh?(y/n): y
```

# BÀI TẬP

10. Viết chương trình tạo ra một menu tương tác cho phép người dùng lựa chọn tính số pi, số e,  $\sin(x)$ . Sau khi tính xong thì đưa ra câu hỏi xem người dùng có tiếp tục tính tiếp hay không, lặp cho đến khi người dùng muốn thoát. Ví dụ:

```
CHUONG TRINH TINH
```

```
1- So pi
```

```
2- So e
```

```
3- sin(x)
```

```
An phim khac: Thoat
```

```
Moi ban lua chon chuong trinh: 1
```

```
Ban da chon chuong trinh 1
```

```
So pi = 3.14
```

```
Ban co muon tiep tục chuong trinh?(y/n): y
```