



BÀI THỰC HÀNH SỐ 5

1. MỤC TIÊU

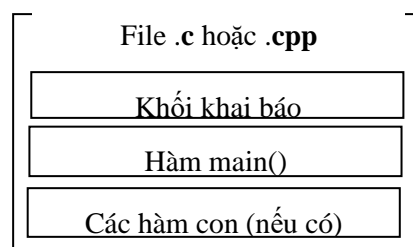
- Hiểu và viết được chương trình với lệnh lặp **while**.
- Viết các hàm do người lập trình tự định nghĩa (**user define function**) với lệnh **while**.
- Hiểu và viết được chương trình với lệnh lặp **for**.

2. LÝ THUYẾT CẦN GHI NHỚ

2.1. Cấu trúc một chương trình có sử dụng hàm trong C.

2.1.1. Cách 1: sử dụng 1 file cho toàn bộ chương trình

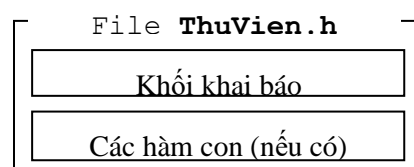
- **Khởi khai báo:** Bao gồm các khai báo về:
 - Khai báo tên các thư viện chuẩn của C được sử dụng trong chương trình.
 - Hằng số sẽ sử dụng trong chương trình.
 - Các kiểu dữ liệu tự định nghĩa.
 - Các biến toàn cục.
 - Hàm con (các nguyên mẫu hàm - prototype).
- **Hàm chính** (**main()**): Chứa các biến, các lệnh và các lời gọi hàm cần thiết trong chương trình.
- **Các hàm con:**
 - Được sắp xếp sao cho mỗi hàm nằm trên 1 đoạn riêng.
 - Không đặt nội dung của hàm này chứa trong hàm khác, hoặc nội dung của 2 hàm có phần giao nhau.
 - Không cần quan tâm thứ tự sắp xếp trước/sau của các hàm.



2.1.2. Cách 2: sử dụng 2 file cho toàn bộ chương trình

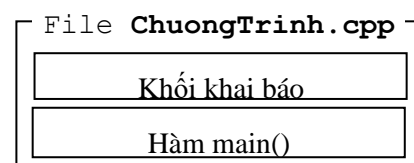
2.1.2.1. File thư viện chứa các hàm do người dùng tự tạo

- **Khởi khai báo:** Bao gồm các khai báo về:
 - Khai báo tên thư viện chuẩn của ngôn ngữ C được sử dụng trong chương trình.
 - Các kiểu dữ liệu tự định nghĩa.
 - Hằng số sẽ sử dụng trong chương trình.
 - Các biến toàn cục.
 - Hàm con (các nguyên mẫu hàm - prototype).
- **Các hàm con:** (tương tự như cách 1)



2.1.2.2. File chương trình

- **Khởi khai báo:** khai báo tên file thư viện chứa các hàm do người dùng tự tạo. Tên file (kể cả phần mở rộng) được đặt trong dấu nháy đôi.
- **Hàm chính** (**main()**): Chứa các biến, các lệnh và các lời gọi hàm cần thiết trong chương trình.





2.2. Cấu trúc for

2.2.1. Công dụng

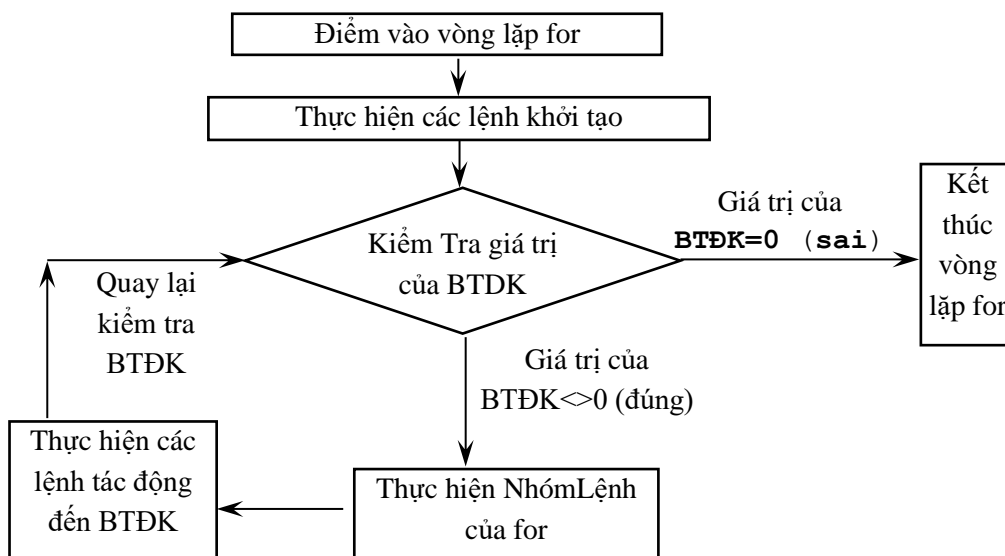
Cấu trúc for thường được dùng trong các trường hợp số lần lặp của vòng lặp được biết trước (giá trị cố định).

2.2.2. Cú pháp

```
for (lệnh khởi tạo; btđk; lệnh tác động đến điều kiện lặp)
{
    NhómLệnh;
}
```

Các thành phần trong ngoặc của for đều có thể vắng mặt tuy nhiên phải để đầy đủ các dấu chấm phẩy (;).

2.2.3. Lưu đồ hoạt động của cấu trúc for



3. BÀI TẬP THỰC HÀNH CƠ BẢN

Tổ chức chương trình dưới dạng các hàm chức năng cho các bài tập sau:

Bài 1. *Số may mắn:* giả sử 1 số được gọi là số may mắn nếu số đó chỉ chứa toàn các số 6 hoặc số 8. Viết chương trình cho nhập số nguyên n, xét xem n có là số may mắn hay không?

Ví dụ $n=686 \Rightarrow 686$ là số may mắn.

$n=68626 \Rightarrow 686\underline{2}6$ KHÔNG phải số may mắn.

Hướng dẫn

- Viết hàm, trong hàm sử dụng vòng lặp while để trích xuất từng chữ số trong số vừa nhập.
- Kiểm tra chữ số vừa trích xuất đó có phải là số 6 hoặc 8 không



```
void kiem_tra_so_may_man(int so){
    int dem =0;
    while(so >0) {
        // su dung phép % và / để lấy từng chữ số trong so
        // kiem tra từng chữ số này khác 8 hoặc 6 thì tăng biến dem
    }
    // neu bien dem >0, in ra ket qua day khong phai la so may man
}
```

Bài 2. Viết chương trình nhập vào số nguyên n gồm ba chữ số. Xuất ra màn hình theo thứ tự đảo ngược của các chữ số.

Ví dụ n=291. Xuất ra 192.

Hướng dẫn

- Tương tự bài 1, sử dụng vòng lặp while và phép %, / để xuất ra từng chữ số

Bài 3. $S = 1 + 2 + 3 + \dots + n$

Hướng dẫn

- Khai báo biến tổng S và gán giá trị ban đầu là 0
- Sử dụng vòng lặp for

```
void tinh_tong(int n){
    int S =0;
    for(int i=1; i<=n; i++){
        S+=i;
    }
    // in ra kết quả
}
```



Bài 4. $S = 1 + 3 + 5 + \dots + (2n+1)$

Hướng dẫn

- Khai báo biến tổng S và gán giá trị ban đầu là 0
- Sử dụng vòng lặp for với biến i từ 1 đến $2*n + 1$
- Kiểm tra nếu i là số lẻ thì cộng

Bài 5.

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

Hướng dẫn

- Tương tự như bài 4

$$S = \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n+1}$$

Bài 6.

$$S = \frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \dots + \frac{1}{2n}$$

Bài 7.

$$S = 1 + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{2n-1}$$

Bài 8.

$$S = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}$$

4. BÀI TẬP NÂNG CAO

Bài 1. Cho nhập số nguyên dương n. Kiểm tra xem các chữ số của n có toàn lẻ (hay toàn chẵn) không?

Bài 2. Cho nhập số nguyên dương n ($100 \leq n < 1000$). Hãy in ra vị trí chứa số lớn nhất. Ví dụ nhập 579, in ra: “số lớn nhất ở hàng đơn vị”

Bài 3. Viết chương trình nhập vào số nguyên n gồm ba chữ số. Xuất ra màn hình theo thứ tự tăng dần của các chữ số.

Ví dụ n=291. Xuất ra 129.



Bài 4. Cho nhập số nguyên dương n. Kiểm tra các chữ của số nguyên dương n có giảm dần từ trái sang phải hay không?

Bài 5.

$$S = \frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \dots + \frac{2n+1}{2n+2}$$

Bài 6. $P = 1 \times 2 \times 3 \times \dots \times n$

Bài 7. $P = 1 \times 3 \times 5 \times \dots \times (2n-1)$

Bài 8. $P = k^n$ (không dùng hàm pow, chỉ dùng các lệnh lặp để tính)

Bài 9. $S = (1) + (1+2) + (1+2+3) + (1+2+3+4) + (1+2+3+4+5) + \dots + (1+2+3+\dots+n)$

Bài 10. $S = (1) + (1 \times 2) + (1 \times 2 \times 3) + \dots + (1 \times 2 \times 3 \times \dots \times n)$
