

# BÀI THỰC HÀNH 1

## Nội dung:

- Phong cách lập trình, tối ưu mã lệnh
- Hàm main có đối số, hàm có đối số mặc định
- Macro, hàm inline, chồng hàm

### Bài 1.1. Viết lại mã lệnh cho rõ ràng, dễ đọc hơn

a. Viết lại chương trình sau:

```
#include <stdio.h>
int main(){int a,b,c;a=5;b=10;c=a+b;printf("%d",c);return 0;}
```

b. Viết lại chương trình sau:

```
#include <stdio.h>
int main() {
    int d;
    scanf("%d", &d);
    if (d >= 0) {
        if (d < 50) printf("Truot");
        else {
            if (d < 65) printf("Trung binh");
            else {
                if (d < 80) printf("Kha");
                else {
                    printf("Gioi");
                }
            }
        }
    }
    return 0;
}
```

c. Đoạn mã sau in bảng cửu chương từ 2 đến 9 nhưng lặp code nhiều lần. Hãy viết lại gọn gàng và dễ hiểu hơn.

```
#include <stdio.h>
int main() {
    int i;
    for (i = 1; i <= 10; i++) printf("2 x %d = %d\n", i, 2*i);
    for (i = 1; i <= 10; i++) printf("3 x %d = %d\n", i, 3*i);
    for (i = 1; i <= 10; i++) printf("4 x %d = %d\n", i, 4*i);
    for (i = 1; i <= 10; i++) printf("5 x %d = %d\n", i, 5*i);
    for (i = 1; i <= 10; i++) printf("6 x %d = %d\n", i, 6*i);
    for (i = 1; i <= 10; i++) printf("7 x %d = %d\n", i, 7*i);
    for (i = 1; i <= 10; i++) printf("8 x %d = %d\n", i, 8*i);
    for (i = 1; i <= 10; i++) printf("9 x %d = %d\n", i, 9*i);
    return 0;
}
```

## Bài 1.2 Tối ưu mã lệnh

a. Tối ưu tính toán tổng dãy số

```
#include <stdio.h>

int main() {
    //nhập số n
    long sum = 0;
    for (int i = 1; i <= n; i++) {
        sum += i;
    }
    printf("Tong = %ld\n", sum);
    return 0;
}
```

b. Tối ưu vòng lặp tính tổng các số chẵn

```
#include <stdio.h>
#define MAX 100000

int main() {
    int n, sum = 0;
    do{
        printf("nhap 1 <= n <= %d: ",MAX);
        scanf("%d",&n);
    }while (n < 1 || n > MAX);

    // Phiên bản chưa tối ưu: lặp qua tất cả các số
    for (int i = 1; i <= N; i++)
        if (i % 2 == 0)
            sum += i;

    printf("Tong cac so chan tu 1 den %d la: %d\n", N, sum);
    return 0;
}
```

## Bài 1.3 Dùng hàm main có đối số

- Viết chương trình có tên Cal sao cho khi gõ: Cal 250 - 170 sẽ xuất ra màn hình  
 $250 - 170 = 80$  (có thể thay dấu – bằng +, \*, /)
- Viết chương trình có tên Avg sao cho khi gõ: Avg 3 2.5 7 15 sẽ xuất ra màn hình Avg  
of Array: 6.875

## Bài 1.4 Viết chương trình thực hiện:

- Viết và sử dụng macro để hoán đổi giá trị giữa 2 phân tử a, b;
- Viết một macro để in ra một dãy số gồm n phân tử ra màn hình;

- c) Viết macro tìm số nhỏ nhất trong 2 số và áp dụng macro này để tìm số nhỏ nhất trong một dãy số của yêu cầu b;
- d) Viết hàm main() nhập các giá trị cần thiết và gọi lần lượt các macro trên thực hiện để kiểm tra kết quả.

**Bài 1.5** Viết các hàm thực hiện các yêu cầu sau:

- a) Hàm nội tuyến để tìm số lớn nhất trong ba số nguyên a, b, c;
- b) Viết hàm tính học phí, tham số nhận vào là số tín chỉ và đơn giá mỗi tín chỉ (mặc định là 550.000 VNĐ), trả về học phí được tính theo công thức: số tín chỉ \* đơn giá mỗi tín chỉ;
- c) Viết hàm tính học bổng, tham số nhận vào là hệ số (mặc định là 1.0), mức học bổng cơ bản (mặc định là 600.000), trả về học bổng của là: mức học bổng cơ bản \* hệ số;  
Nhập vào điểm trung bình, sử dụng hàm tính học bổng để tính học bổng cho sinh viên dựa trên điểm trung bình, biết:
  - Nếu  $7 \leq \text{điểm trung bình} < 8$ : hệ số 1.0, mức học bổng 600.000
  - Nếu  $8 \leq \text{điểm trung bình} < 9$ : hệ số 1.3, mức học bổng 600.000
  - Nếu  $9 \leq \text{điểm trung bình} \leq 10$ : hệ số 1.5, mức học bổng 600.000
- d) Viết và sử dụng chồng hàm để tính diện tích của hình chữ nhật và hình tròn.
- e) Hàm main() nhập các giá trị cần thiết và gọi lần lượt các hàm trên thực hiện để kiểm tra kết quả.

**Bài 1.6\*** Viết các hàm (chọn giải thuật tốt) thực hiện các yêu cầu sau:

- a) Kiểm tra số nguyên tố, biết số nguyên tố chỉ chia hết cho 1 và chính nó.
- b) Tìm ước chung lớn nhất của 2 số nguyên.
- c) Kiểm tra một số có là số armstrong hay không? Số armstrong là số có giá trị bằng tổng lập phương các chữ số của chính nó.  
Ví dụ: 153 là một số armstrong bởi vì:  

$$153 = (1)^3 + (5)^3 + (3)^3 = 1 + 125 + 27 = 153$$
- d) Hàm main() nhập các giá trị và gọi các hàm trên thực hiện để kiểm tra kết quả.

**Bài 1.7\*** Viết chương trình thực hiện các yêu cầu sau:

- a) Viết một macro In\_Range(x, min, max) để kiểm tra xem giá trị x có nằm trong khoảng [min, max] hay không;
- b) Nhập vào một dãy số gồm n phần tử là số thực ( $2 < n < 10$ ), sử dụng macro ở yêu cầu a để ràng buộc giá trị n;
- c) Viết macro Arr\_Min(arr, n, kq) để tìm giá trị nhỏ nhất trong một mảng arr có n phần tử và gán kết quả cho kq.
- d) Hàm main() nhập các giá trị và gọi các hàm, macro trên thực hiện để kiểm tra kết quả.

**Bài 1.8\*** Cho biết kết quả chạy các chương trình sau:

a)

```
#include<stdio.h>
#define Square(x) (x)*(x)

int main() {
    int a=2;
    printf("%d",Square(a));
}
```

b)

```
#include<stdio.h>
#define Square(x) x*x

int main() {
    int a=2;
    printf("%d",Square(a+3));
}
```

c)

```
#include<stdio.h>
#define Square(x) (x)*(x)

int main() {
    int a=2;
    printf("%d",Square(a+3));
}
```

----- hết -----