BUỔI 4: HÀM -- FULL HOUSE

I. Định nghĩa về hàm:

a. Là một **nhóm các câu lệnh** cùng nhau thực hiện một hay một số chức năng nào đó.
b. Một số chức năng: cộng, trừ, nhân, chia, tìm max, tìm min, kiểm tra số thỏa mãn,….

II. Phân loại

a. Hàm có sẵn: pow, sqrt, abs,….
b. Tự định nghĩa: (do người lập trình viết ra)
   i. Có kiểu dữ liệu trả về: int, float,…

```
3  int main(){
4
5      return 0;
6  }
```

```
float thuong(float a, int b){
    return a/b;
}
```

   ii. Không có kiểu dữ liệu trả về: void

```
3  void max(int a, int b){
4      if(a>b) printf("%d",a);
5      else printf("%d",b);
6  }
```

III. Mục đích (chia nhỏ)

a. Dễ quản lí, dễ debug
b. Rõ ràng, tường minh
c. Tái sử dụng.

IV. Cấu trúc

a. Hàm có sẵn: pow, printf, scanf,…
b. Tự định nghĩa:
    i. Khai báo: Chỉ khai báo nguyên mẫu

Kieu_Tra_Ve Ten_Ham (Danh_sach_tham_so);

```
8  int tong(int a, int b);
9
10 bool check(float a, char b, int c);
11
```

    ii. Định nghĩa: Viết nội dung cụ thể.

Kieu_Tra_Ve Ten_Ham (Danh_sach_tham_so) {
**   //Khối lênh**
}

```
3  int main(){
4
5      return 0;
6  }
```

```
3  float tong(float a, int b){
4      return a/b;
5  }
```

```
3  void max(int a, int b){
4      if(a>b) printf("%d",a);
5      else printf("%d",b);
6  }
```

***Kiểu trả về**: int, float, void, kiểu con trỏ,....*
***Tên hàm**: Lập trình viên tự đặt, tránh các keyword,...*
***Danh sách tham số:** các biến cần thiết để hàm hoạt động.*

V. Gọi hàm

Để gọi hàm, bạn đơn giản cần truyền các tham số được yêu cầu cùng với tên của hàm và nếu hàm trả về các giá trị, bạn có thể dự trữ các giá trị trả về này

VI. Một số ví dụ:

a. Gọi hàm chao

```c
#include <stdio.h>

void chao(){
    printf("Xin chao");
}

int main(){
    chao();
}
```

```c
#include <stdio.h>

void chao(){
    printf("\nXin chao");
}

int main(){
    for(int i=1;i<=10;++i){
        chao();
    }
}
```

b. Tìm số lớn nhất trong 2 số

```c
#include <stdio.h>

int max(int a, int b){
    if(a<b) return b;
    else return a;
}

int main(){
    int c = max(10,30);
    printf("%d",c);
    return 0;
}
```

c. Viết hàm tính tổng các số không vượt quá n.

```c
#include <stdio.h>

long long tong(int n){
    return 1ll*n*(n+1)/2;
}

int main(){
    int n;
    scanf("%d",&n);
    long long x= tong(n);
    printf("%lld",x);
}
```

d. Kiểm tra số nguyên tố (Bài 5.04)

```c
#include <stdio.h>

int check(int n){
    if(n<2)  return 0;
    for(int i=2;i<n;++i){
        if(n%i==0)  return 0;
    }
    return 1;
}

int main(){
    int a;
    scanf("%d",&a);
    if(check(a)==1)
        printf("Day la so nguyen to");
    else printf("Day khong phai la so nguyen to");
}

```

e. Kiểm tra tổng chữ số n có phải là số nguyên tố hay không?

```c
#include <stdio.h>

int tong(int n){
    int s=0;
    while(n){
        s+=n%10;
        n/=10;
    }
    return s;
}

int check(int n){
    if(n<2)  return 0;
    for(int i=2;i<n;++i){
        if(n%i==0)  return 0;
    }
    return 1;
}

int main(){
    int a;
    scanf("%d",&a);
    int x= tong(a);
    if(check(x)==1)
        printf("YES");
    else printf("NO");
}

```

VII. Tham trị và tham chiếu

a. Tham trị

i. Chúng ta vẫn thường dùng hàm trong hầu hết các bài học trước và truyền tham số theo cách truyền theo tham trị.

Ví dụ: **void tinh(int x){}.**

```
3  void tinh(int x){
4      x += 10;
5  }
```

ii. Khi một biến **a** được truyền vào lời gọi hàm **tinh(int x)** làm tham số dưới dạng **tham trị**, thì biến **x** của hàm **tinh(int x)** và biến **a** là **hai biến độc lập**. Bởi vì khi tham số của hàm **tinh(int x)** là **tham trị**, hàm này sẽ tạo ra một biến mới và **sao chép giá trị** của **a** vào. Do đó, nếu hàm mà **thay đổi giá trị của x trong hàm** tinh(int x) này thì **không tác động** gì tới giá trị của biến a.

```
1   #include <stdio.h>
2
3   void tinh(int x){
4       x += 10;
5   }
6
7   int main(){
8       int a=5;
9       tinh(a);
10      printf("%d",a);
11  }
```

b. Tham chiếu

i. Khai báo hàm sử dụng tham số dưới dạng tham chiếu bằng cách thêm dấu **&** vào trước tham số đó

Ví dụ: **void tinh(int &x){}.**

```
void tinh(int &x){
    x += 10;
}
```

ii. Khi một biến a được truyền vào lời gọi hàm **tinh(int &x)** làm tham số dưới dạng **tham chiếu**, thì biến **x** của hàm **tinh(int &x)** và biến **a** thực chất **là một**. Do đó, nếu hàm mà thay đổi giá trị của x trong hàm **tinh(int &x)** này thì đồng nghĩa tại nơi gọi hàm biến a cũng **bị thay đổi theo**.

iii. Nếu tham số là mảng hoặc chuỗi, thì tham số này được truyền theo tham chiếu.

```
1  #include <stdio.h>
2
3  void tinh(int &x){
4      x += 10;
5  }
6
7  int main(){
8      int a=5;
9      tinh(a);
10     printf("%d",a);
11 }
```