

## NỘI DUNG CHÍNH BUỔI 8

# HÀM

### (1) Khái niệm

- Hàm là một đoạn chương trình độc lập nhằm thực hiện trọn vẹn một công việc nhất định.  
=> Tính đóng gói giúp an toàn dữ liệu (người dùng chỉ biết input / output của hàm, không can thiệp được khối xử lý bên trong thân hàm).
- Các hàm là những đoạn chương trình nhỏ giúp chia nhỏ một vấn đề lớn  
=> Làm cho vấn đề lớn (bài toán, chương trình) trở nên đơn giản vì được phân hoạch, giải quyết từng phần trong các hàm con nhỏ hơn.  
=> Dễ quản lý, kiểm lỗi từng phần chương trình.  
=> Có thể tái sử dụng các hàm để kế thừa, nâng cấp, phát triển thành các chương trình lớn hơn.

### (2) Cú pháp khai báo

```
<kiểu_dữ_liệu_hàm> <tên_hàm> (<kiểu_dữ_liệu_1> <tên_tham_số_hình_thức_1>, <kiểu_dữ_liệu_2> <tên_tham_số_hình_thức_2>, ...)  
{  
    //Khởi lệnh xử lý;  
    return giá_trị; //chỉ dùng cho hàm có kiểu khác void  
}
```

Trong đó:

- **<kiểu\_dữ\_liệu\_hàm>** gồm 2 loại:
  - **void** (=> không trả về 1 giá trị nào)
  - **khác void** (VD: **int**, **float**, **double**, **char**, **bool**, ... => trả về 1 giá trị tương ứng với kiểu)
- **<tên\_hàm>**, **<tên\_tham\_số\_hình\_thức\_i>** thỏa quy tắc **3 KHÔNG** tương tự như cách đặt tên biến (không chứa khoảng trắng, không chứa ký tự đặc biệt, không bắt đầu là số).
- Lệnh **return giá\_trị; chỉ dùng cho hàm có kiểu khác void** để trả về giá trị cho nơi gọi hàm.

**Ví dụ 1: Xét yêu cầu:**

- ⇒ **Viết hàm tìm số lớn hơn trong 2 số nguyên nhập vào**
- ⇒ **Viết hàm tìm số lớn nhất trong 3 số nguyên nhập vào**
- ⇒ **Viết hàm tìm số lớn nhất trong 4 số nguyên nhập vào**
- ⇒ ...
- ⇒ **Viết hàm tìm số lớn nhất trong n số nguyên nhập vào**

**Gợi ý:**

- Nhập 2 số nguyên: (2, 7) => Max\_2\_so: 7
- Nhập 3 số nguyên: (2, 7, 5) => Tìm max của 2 số 2, 7 trước (=>Max\_2\_so: 7), sau đó tìm max của Max\_2\_so với số 5 còn lại (=>Max\_3-so: 7)
- Nhập 4 số nguyên: (2, 7, 5, 9) => Tìm max của 2 số 2, 7 trước (=>Max\_2\_so: 7), sau đó tìm max của Max\_2\_so với số 5 (=>Max\_3-so: 7), cuối cùng tìm max của Max\_3\_so với số 9 còn lại (=>Max\_4-so: 9)
- ...
- ⇒ **Viết hàm tìm số lớn hơn trong 2 số nguyên nhập vào. Sau đó, tái sử dụng hàm này trong các hàm khác.**

```
//Cách 1: Dùng hàm if để so sánh và trả về giá trị lớn hơn trong 2 số
int Max2So(int so1, int so2)
{
    if(so1 > so2)
        return so1;
    else
        return so2;
}
```

```
/* Cách 2: Khai báo thêm biến max để nhận giá trị lớn hơn trong 2 số và trả về giá trị của biến max này*/
```

```
int Max2So(int so1, int so2)
{
    int max;
    if(so1 > so2)
        max = so1;
    else
        max = so2;
    return max;
}
```

```
/* Cách 3: Dùng toán tử biểu thức điều kiện để so sánh giá trị của 2 số và trả về giá trị số lớn hơn => xem thêm lý thuyết chương 2, slide 23, 24 */
```

```
int Max2So(int so1, int so2)
{
    return (so1 > so2) ? so1 : so2; //toán tử biểu thức điều kiện => Xem chương 2 - slide 23, 24
}
```

### (3) Lời gọi hàm

- *Hàm kiểu void:*

**tên\_hàm**(tên\_biến\_1, tên\_biến\_2, ...);

- *Hàm kiểu khác void:*

- In giá trị ra màn hình:

<kiểu\_dữ\_liệu\_i> <tên\_biến\_i>; //Khai báo biến i có kiểu trùng với <kiểu\_dữ\_liệu\_hàm> để nhận giá trị trả về của hàm

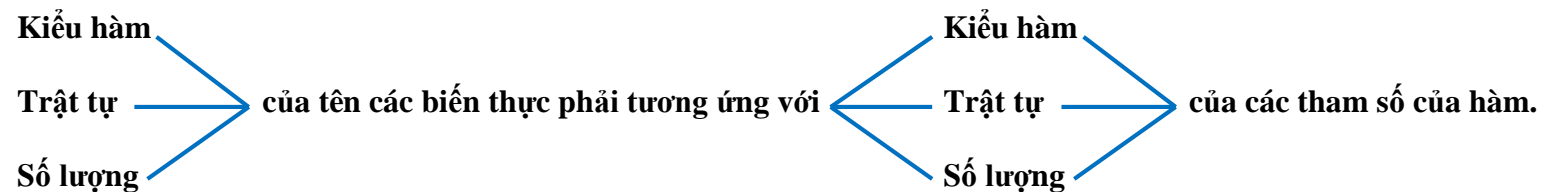
<tên\_biến\_i> = **tên\_hàm**(tên\_biến\_1, tên\_biến\_2, ...); //Gán giá trị cho biến i = giá trị trả về của hàm

//Xuất giá trị biến i ra màn hình

- Sử dụng trong hàm if:

```
if( tên_hàm(tên_biến_1, tên_biến_2, ...) <toán_tử_quan_hệ> <giá_trị_i>) /*So sánh giá trị trả về của hàm với giá trị i
                                     (> >= < <= == !=)*
{
    ...
}
```

❖ Lưu ý:



**Ví dụ:** Viết hàm main gọi hàm tìm số lớn hơn trong 2, 3, 4, ... số nguyên nhập vào:

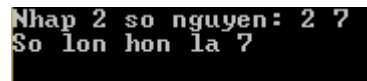
```
void main()
{
    int a,b;
    printf("Nhập 2 số nguyên: ");
    scanf("%d%d",&a,&b);
    int m = Max2So(a,b); //Khai báo biến nguyên m nhận giá trị trả về của hàm Max2So(a,b)
    printf("Số lớn hơn là %d\n",m); //In giá trị m
    _getch();
}
```

⇒ **Debug (F10) CT và xem kết quả**

```
Nhập 2 số nguyên: 2 7
Số lớn hơn là 7
```

```
//Hoặc:
void main()
{
    int a,b;
    printf("Nhap 2 so nguyen: ");
    scanf("%d%d",&a,&b);
    printf("So lon hon la %d\n",Max2So(a,b)); //In trực tiếp giá trị trả về của hàm Max2So(a,b)
    _getch();
}
```

⇒ **Debug (F10) CT và xem kết quả**



```
Nhap 2 so nguyen: 2 ?
So lon hon la ?
```

```
void main()
{
    int a,b,c;
    printf("Nhap 3 so nguyen: ");
    scanf("%d%d%d",&a,&b,&c);
    int m2=Max2So(a,b); /*Khai báo và gán giá trị cho biến nguyên m2 là giá trị lớn hơn trong
                        2 số a, b*/
    int m3=Max2So(m2,c); /*Khai báo và gán giá trị cho biến nguyên m3 là giá trị lớn hơn trong
                        2 số m2, c => số lớn nhất */
    printf("So lon nhat la %d\n",m3);
    _getch();
}
```

```
//Hoặc:
void main()
{
    int a,b,c;
    printf("Nhap 3 so nguyen: ");
    scanf("%d%d%d",&a,&b,&c);
    printf("So lon nhat la %d\n",Max2So(Max2So(a,b),c)); /*In trực tiếp giá trị trả về của hàm
                                                         lồng hàm Max2So */
    _getch();
}
```

⇒ **Debug (F10) CT và xem kết quả**

```
Nhap 3 so nguyen: 2 7 5
So lon nhat la ?
```

```
void main()
{
    int a,b,c,d;
    printf("Nhap 4 so nguyen: ");
    scanf("%d%d%d%d",&a,&b,&c,&d);
    int m2=Max2So(a,b);
    int m3=Max2So(m2,c);
    int m4=Max2So(m3,d);
    printf("So lon nhat la %d\n",m4);
    _getch();
}
```

Hoặc:

```
int max_ab=Max2So(a,b);
int max_cd=Max2So(c,d);
int max_abcd=Max2So(max_ab,max_cd);

printf("So lon nhat la %d\n",max_abcd);
```

//Hoặc:

```
void main()
{
    int a,b,c,d;
    printf("Nhap 4 so nguyen: ");
    scanf("%d%d%d%d",&a,&b,&c,&d);
    printf("So lon nhat la %d\n",Max2So(Max2So(Max2So(a,b),c),d)); //Hàm lồng hàm
    _getch();
}
```

Hoặc:

```
printf("So lon nhat la %d\n",Max2So(Max2So(a,b),Max2So(c,d)));
```

⇒ **Debug (F10) CT và xem kết quả**

```
Nhap 4 so nguyen: 2 7 5 9
So lon nhat la 9
```

**Ví dụ 2: Viết CT giải phương trình bậc 2 sử dụng hàm**

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
//Ham giai va bien luan phuong trinh bac 1: ax+b=0
void GiaiPTBac1(double a,double b) //input: 2 tham so thuc la 2 he so a, b
{
    if(a==0)
        if(b==0)
            printf("PTVSN");
        else
            printf("PTVN");
    else
        printf("PT co nghiem x = %lf",-b/a);
}

//Ham giai va bien luan phuong trinh bac 2: ax2+bx+c=0
void GiaiPTBac2(double a, double b, double c) //input: 3 tham so thuc la 3 he so a, b, c
{
    if(a==0)
        GiaiPTBac1(b,c); //Goi ham GiaiPTBac1 bx+c=0 (2 he so la b, c)
    else
    {
        double delta = b*b-4*a*c;
        if(delta<0)
            printf("PTVN vi delta<0");
        else if(delta==0)
            printf("PT co nghiem kep x1 = x2 = %lf",-b/(2*a));
        else
        {
            double x1 = (-b-sqrt(delta))/(2*a), x2 = (-b+sqrt(delta))/(2*a);
            printf("PT co 2 nghiem phan biet: x1 = %lf, x2 = %lf",x1,x2);
        }
    }
}
```

```
void main()
{
    //Khai bao 3 bien luu gia tri 3 he so a,b,c
    double hsa,hsb,hsc;

    //Nhap lieu cho 3 he so a,b,c
    printf("Nhap 3 he so a, b, c: ");
    scanf("%lf%lf%lf",&hsa,&hsb,&hsc);

    /*Goi ham GiaiPTBac2 truyen 3 bien thuc la hsa, hsb, hsc tuong ung voi 3 tham so hinh thuc
    a, b, c cua ham*/
    GiaiPTBac2(hsa,hsb,hsc);
    _getch();
}
```

**Ví dụ 3: Làm lại ví dụ in 3 loại bảng cửu chương: n, m=>n, tổng hợp (xây dựng 3 hàm con)**

```
//Ham in bang cuu chuong thu n
void In_BCC_n()
{
    int n;
    do
    {
        printf("Nhap n tu 2 => 9: ");
        scanf("%d",&n);
    }while(n<2||n>9);
    for(int i=1;i<=10;i++)
        printf("%d x %2d = %2d\n",n,i,n*i);
}
```



```
//Ham in cac bang cuu chuong tu m => n
void In_BCC_mn()
{
    int m,n;
    do
    {
        printf("Nhap m, n tu 2 => 9, m<=n, n-m<=4: ");
        scanf("%d%d", &m, &n);
    }while(m<2 || m>9 || n<2 || n>9 || m>n || n-m>4);
    for(int d=1; d<=10; d++)
    {
        for(int c=m; c<=n; c++)
            printf("%d x %d = %d\t", c, d, c*d);
        printf("\n");
    }
}

//Ham in bang cuu chuong tong hop
void In_BCC_th()
{
    for(int d=1; d<=10; d++)
    {
        for(int c=1; c<=10; c++)
            printf("%3d\t", c*d);
        printf("\n\n");
    }
}

void main()
{
    In_BCC_n(); //Goi ham in bang cuu chuong thu n
    In_BCC_mn(); //Goi ham in cac bang cuu chuong tu m => n
    In_BCC_th(); //Goi ham in bang cuu chuong tong hop
    _getch();
}
```

⇒ **Debug (F10) CT và xem kết quả, lưu ý nhấn F11 để nhảy vào các hàm con.**

**(4) Nguyên mẫu hàm**

- Về nguyên tắc khi gọi một hàm thì hàm đó phải được khai báo trước, nếu không chương trình sẽ báo lỗi. Tuy nhiên cũng có thể gọi một hàm chưa được khai báo trước bằng cách khai báo trước **nguyên mẫu hàm**.
- Nguyên mẫu hàm là dòng đầu của hàm có thêm dấu ;**

**Ví dụ 3: Viết lại CT in 3 loại bảng cửu chương, khai báo nguyên mẫu hàm**

```
//Khai báo thư viện
void In_BCC_n(); //Khai báo nguyên mẫu hàm in bang cuu chuong thu n
void In_BCC_mn(); //Khai báo nguyên mẫu hàm in cac bang cuu chuong tu m => n
void In_BCC_th(); //Khai báo nguyên mẫu hàm in bang cuu chuong tong hop
void main()
{
    In_BCC_n(); //Goi ham in bang cuu chuong thu n
    In_BCC_mn(); //Goi ham in cac bang cuu chuong tu m => n
    In_BCC_th(); //Goi ham in bang cuu chuong tong hop
    _getch();
}
//Ham in bang cuu chuong thu n
void In_BCC_n()
{
    ...
}
//Ham in cac bang cuu chuong tu m => n
void In_BCC_mn()
{
    ...
}
//Ham in cac bang cuu chuong tong hop
void In_BCC_th()
{
    ...
}
```

⇒ **Tạo hệ thống menu chọn lựa cho CT in 3 loại bảng cửu chương:**

```
void main()
{
    int chon;
    do
    {
        system("cls");//xóa màn hình: clear screen => thư viện <stdlib.h>
        printf("-----CHUONG TRINH IN BANG CUU CHUONG-----\n");
        printf("1. In bang cuu chuong thu n\n");
        printf("2. In cac bang cuu chuong tu m den n\n");
        printf("3. In bang cuu chuong tong hop\n");
        printf("0. Thoat CT\n");
        printf("-----\n");
        printf("Ban chon: ");
        scanf("%d",&chon);
        switch(chon)
        {
            case 0: printf("Dang thoat CT...");break;
            case 1: In_BCC_n();break;
            case 2: In_BCC_mn();break;
            case 3: In_BCC_th();break;
            default: printf("Ban da chon sai. Moi chon lai.");
        }
        //_getch();//tạm dừng màn hình xem kết quả, có thể sử dụng lệnh: system("pause");
    }while(chon!=0);
}
```

**(5) Truyền tham số cho hàm****• Truyền tham trị:**

- Giá trị của các tham số thực được truyền cho các tham số hình thức tương ứng của hàm.
- Mọi thay đổi trong hàm trên các tham số hình thức sẽ không ảnh hưởng tới giá trị ban đầu của các tham số thực.

VD:

```
//Hàm hoán vị hai số nguyên
void HoanVi(int a, int b)
{
    int tam; //khai báo biến cục bộ tam
    tam = a; a = b; b = tam;
}
void main()
{
    int x = 3, y = 7;
    HoanVi(x, y); //Lời gọi hàm HoanVi với hai tham số thực x và y
    <In x, y>; //x và y không thay đổi x = 3, y = 7
}
```

**• Truyền tham trở: (xem lý thuyết chương 10)**

- Địa chỉ của các tham số thực (có dấu & đặt trước) được truyền cho các tham số hình thức tương ứng của hàm và các tham số hình thức phải được khai báo dưới dạng con trỏ (có dấu \* đặt trước).
- Bên trong hàm có thể truy nhập trực tiếp đến vùng nhớ các tham số thực thông qua các con trỏ này.

VD:

```
//Hàm hoán vị hai số nguyên
void HoanVi(int *a, int *b)
{
    int tam; //khai báo biến cục bộ tam
    tam = *a; *a = *b; *b = tam;
}

void main()
{
    int x = 3, y = 7;
    HoanVi(&x, &y); //Lời gọi hàm HoanVi với hai tham số thực &x và &y
    <In x, y>; // x và y đã bị thay đổi x = 7, y = 3
}
```

- **Truyền tham chiếu (tham biến):**

- Tham số hình thức cũng chính là tham số thực (tham số hình thức được lưu tại địa chỉ lưu tham số thực, dùng chung vùng nhớ với tham số thực).
- Mọi thay đổi trong hàm trên các tham số hình thức sẽ **đồng thời làm thay đổi** giá trị ban đầu của của các tham số thực.

```
//Hàm hoán vị hai số nguyên
void HoanVi(int &a, int &b)
{
    int tam; //khai báo biến cục bộ tam
    tam = a; a = b; b = tam;
}

void main()
{
    int x = 3, y = 7;
    HoanVi(x, y); //Lời gọi hàm HoanVi với hai tham số thực x và y
    <In x, y>; // x và y đã bị thay đổi x = 7, y = 3
}
```

**Ví dụ: Viết hàm nhập giá trị số nguyên**

```
//Truyền tham trị
void NhapSoNguyen(int n)
{
    printf("Nhap so nguyen: ");
    scanf("%d", &n);
}
void main()
{
    int n;
    NhapSoNguyen(n);
    printf("So vua nhap la: %d", n);
    _getch();
}
```

⇒ **CT báo lỗi biến n chưa được gán giá trị khởi động** (vẫn là giá trị rác) vì giá trị của tham số thực n được truyền cho tham số hình thức n theo kiểu tham trị nên dù người dùng đã nhập liệu giá trị cho tham số hình thức n trong hàm NhapSoNguyen thì tham số thực n vẫn không nhận được giá trị tương ứng này. Sửa lại bằng 1 trong 2 cách: truyền tham trở hoặc truyền tham biến như sau:

```
//Truyền tham trở
void NhapSoNguyen(int *pn)
{
    printf("Nhap so nguyen: ");
    scanf("%d", pn);
}
void main()
{
    int n;
    NhapSoNguyen(&n);
    printf("So vua nhap la: %d", n);
    _getch();
}
```

```
//Truyền tham biến
void NhapSoNguyen(int *pn)
{
    printf("Nhap so nguyen: ");
    scanf("%d",pn);
}
void main()
{
    int n;
    NhapSoNguyen(&n);
    printf("So vua nhap la: %d",n);
    _getch();
}
```

**Kết quả:** 