

GV: Nguyễn Văn Sơn

CHƯƠNG 4

HÀM (FUNCTION)

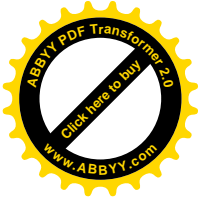
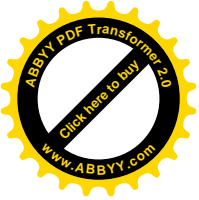
0

GV: Nguyễn Văn Sơn

1. KHÁI NIỆM

- Trong chương trình có một lệnh hoặc khối lệnh nào đó cần lặp đi lặp lại nhiều lần ở những vị trí khác nhau, để không mất thời gian chúng ta tổ chức đoạn lệnh đó thành chương trình con. Khi nào cần dùng ta chỉ việc gọi chương trình con đó để sử dụng.
- Chương trình con đó gọi là hàm (function)

1



GV: Nguyễn Văn Sơn

2. XÂY DỰNG HÀM

[kiểu DL] Tên hàm ([Danh sách tham số hình thức])

```
{  
    [Khai báo biến nội bộ]  
    các câu lệnh;  
    [return [biểu thức]];  
}
```

- Kiểu DL: Nếu hàm phải trả về 1 giá trị thì phải khai báo kiểu dữ liệu trả về. Nếu hàm không trả về giá trị thì dùng kiểu void.

- Danh sách tham số: đây là tham số hình thức hay còn gọi là biến hình thức. Những tham số này sẽ nhận giá trị thực bằng cách truyền tham số mỗi khi hàm được gọi.

- Thân hàm: được giới hạn bởi cặp dấu móc nhọn {...}

2

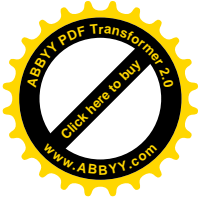
GV: Nguyễn Văn Sơn

3. SỬ DỤNG HÀM

- Sau khi **đã tạo lập** các hàm, **ở chỗ** nào **đó** trong chương trình cần **sử dụng** hàm ta có thể thực hiện bằng cách gọi hàm theo cú pháp sau:

Tên hàm ([danh sách các tham số thực])

3



GV: Nguyễn Văn Sơn

3. SỬ DỤNG HÀM

```
#include <stdio.h>
```

```
int cong2so(int a, int b)
```

```
{
```

```
    return (a+b);
```

```
}
```

```
main()
```

```
{ int x, y, z;
```

```
    printf("Nhap x va y :"); scanf("%d%d",&x&&b);
```

```
    z = cong2so(x,y);
```

```
    printf("Ket qua x + y = %d",z);
```

```
}
```

Xây dựng hàm.
a và b là tham số
hình thức

Gọi hàm,
x và y là tham
số thực được
truyền vào

4

GV: Nguyễn Văn Sơn

4. CÁCH THỨC TRUYỀN THAM SỐ

- **Đối với** các hàm có tham số hình thức thì khi thực hiện phải được truyền giá trị thực cho các tham số này.

- Có 2 cách truyền tham số:

+ Truyền bằng trị: khi tham số hình thức thay đổi thì tham số thực vẫn không thay đổi

+ Truyền bằng biến: Khi tham số hình thức thay đổi thì tham số thực cũng thay đổi theo

5

GV: Nguyễn Văn Sơn

VÍ DỤ VỀ TRUYỀN BẰNG TRỊ VÀ TRUYỀN BẰNG BIẾN

Truyền bằng trị

```
#include <stdio.h>
#include <conio.h>
void hoanvi(int a, int b)
{
    int c;
    c = a;
    a = b;
    b = c;
}
main()
{
    int x=1, y = 2;
    hoanvi(x,y);
    printf("x = %d, y = %d", x,y);
    getch();
}
```

Kết quả: x=1, y=2

Truyền bằng biến

```
#include <stdio.h>
#include <conio.h>
void hoanvi(int &a, int &b)
{
    int c;
    c = a;
    a = b;
    b = c;
}
main()
{
    int x=1, y = 2;
    hoanvi(x,y);
    printf("x = %d, y = %d", x,y);
    getch();
}
```

Truyền bằng biến
thêm dấu &

Kết quả: x=2, y=1

6

GV: Nguyễn Văn Sơn

THAM SỐ THỰC LÀ HẰNG SỐ

Truyền bằng trị

```
#include <stdio.h>
#include <conio.h>
int nhan2so(int a, int b)
{
    return (a*b);
}
int cong2so(int a, int b)
{
    return (a+b);
}
main()
{
    int x, y;
    printf("2*3=%d\n", nhan2so(2,3));
    printf("2+3=%d", cong2so(2,3));
    getch();
}
```

Truyền bằng biến

```
#include <stdio.h>
#include <conio.h>
int nhan2so(int &a, int &b)
{
    return (a*b);
}
int cong2so(int &a, int &b)
{
    return (a+b);
}
main()
{
    int x, y;
    printf("2*3=%d\n", nhan2so(2,3));
    printf("2+3=%d", cong2so(2,3));
    getch();
}
```

**Báo
lỗi**

7

GV: Nguyễn Văn Sơn

5. KHAI BÁO HÀM

Không khai báo hàm

```
#include <stdio.h>
#include <conio.h>
int nhan2so(int a, int b)
{
    return (a*b);
}
int cong2so(int a, int b)
{
    return (a+b);
}

main()
{
    int x, y;
    printf("Nhap x, y:"); scanf("%d%d",&x,&y);
    printf("x*y=%d\n",nhan2so(x,y));
    printf("x+y=%d",cong2so(x,y));
    getch();
}
```

Khai báo hàm (prototype)

```
#include <stdio.h>
#include <conio.h>
int nhan2so(int a, int b);
int cong2so(int a, int b);
main()
{
    int x, y;
    printf("Nhap x, y:"); scanf("%d%d",&x,&y);
    printf("x*y=%d\n",nhan2so(x,y));
    printf("x+y=%d",cong2so(x,y));
    getch();
}
int nhan2so(int a, int b)
{
    return (a*b);
}
int cong2so(int a, int b)
{
    return (a+b);
}
```

8

GV: Nguyễn Văn Sơn

6. HÀM KHÔNG TRẢ VỀ GIÁ TRỊ VÀ HÀM TRẢ VỀ GIÁ TRỊ

```
#include <stdio.h>
#include <conio.h>
void ABC(int n)
{
    for (int i=0; i<n; i++)
        printf("Hoa Sen\n");
}

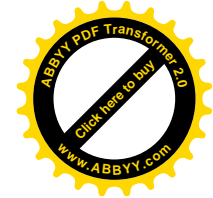
float thuong2so(float a, float b)
{
    return (a/b);
}

main()
{
    ABC(5);
    float z =float(100, 3);
    printf("z = %.2f",z);
    getch();
}
```

Hàm kg trả về giá trị thì dùng kiểu **void**

Hàm trả về giá trị thì dùng kiểu trả về, dùng từ khoá **return** (biểu thức)

9



GV: Nguyễn Văn Sơn

5. CÂU LỆNH RETURN VÀ EXIT

- Câu lệnh return: Khi gặp câu lệnh này trong hàm, máy tính sẽ chuyển quyền điều khiển ra ngoài, không thực hiện các câu lệnh còn lại trong hàm.
- Câu lệnh exit: chấm dứt thực hiện chương trình, trả điều khiển về hệ điều hành.

10

BÀI TẬP

- 1/ Viết hàm tìm số lớn nhất trong 3 số
- 2/ Viết hàm kiểm tra số nguyên n có phải là số nguyên tố không?
- 3/ Viết hàm tìm USCLN của 2 số nguyên
- 4/ Viết hàm tính n!
- 5/ Viết hàm in n số fibonacci đầu tiên
- 6/ Viết hàm tính:

$$T1 = 1 + 2 + 3 + 4 + \dots + n$$

$$T2 = 1 - 2 + 3 - 4 + \dots + (-1)^n$$

$$T3 = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

$$T4 = 1 + \frac{1}{(1+2)} + \frac{1}{(1+2+3)} + \dots + \frac{1}{(1+2+\dots+n)}$$

$$F1 = 1 + \frac{2^2}{\sqrt{2}} + \frac{3^3}{\sqrt[3]{3}} + \dots + \frac{n^n}{\sqrt[n]{n}}$$

$$F2 = 1 + \frac{2!}{(1+2)} + \frac{3!}{(1+2+3)} + \dots + \frac{n!}{(1+2+\dots+n)}$$

11

HÀM ĐỆ QUY

Tại một vị trí nào đó trong thân của một hàm mà gọi lại chính nó thì gọi là hàm đệ quy.

```
void abc()
```

```
{
```

```
...
```

```
  abc();
```

```
...
```

```
  abc();
```

```
...
```

```
}
```

Gọi lại chính nó

12

XÂY DỰNG HÀM ĐỆ QUY

Hàm đệ quy thường được xây dựng theo thuật toán:

```
if (trường hợp suy biến)
{ trình bày cách giải bài toán }
else
{ gọi đệ quy tới hàm đang lập
  với giá trị khác của tham số }
```

Đệ quy là khả thi nếu đệ quy có chặn (còn gọi là điều kiện biên)

13

VD XÂY DỰNG HÀM ĐỆ QUY

Vd1: Tính $n! = ?$

```
long gt(int n)
{
    if (n==0 || n==1)
        return 1;
    else
        return n * gt(n-1);
}
```

Trường hợp suy biến
là $n=0$ hoặc $n=1$
(điều kiện biên)

14

VD XÂY DỰNG HÀM ĐỆ QUY

Vd2: Viết hàm đệ qui để in ra màn hình 10 hàng chữ: Hoa Sen

```
void hoasen(int n)
{
    if (n==0)
        return;
    else
    { printf("Hoa Sen\n");
      hoasen(n-1); }
}
```

Trường hợp suy biến
là $n=0$
(điều kiện biên)

15

VD XÂY DỰNG HÀM ĐỆ QUY

Vd3: Tìm USC của x và y

```
int usc(int x, int y)
{
    if (x*y==0)
        return x+y;
    else if(x>y)
        return usc(x-y,y);
    else
        return usc(x,y-x);
}
```

Trường hợp suy biến
là $x*y=0$
(điều kiện biên)

16

BÀI TẬP VIẾT HÀM ĐỆ QUY

- 1/ Viết hàm tính tổng $T1 = 1+2+3+...+n$ bằng đệ quy
- 2/ Viết hàm tính tổng $T2 = 1-2+3-4+...+n$ bằng đệ quy
- 3/ Viết hàm tìm giá trị của số fibonacci thứ n bằng đệ quy
- 4/ Viết hàm đệ quy để in bảng cửu chương n bằng đệ quy.
- 6/ Viết hàm đệ quy tính:

$$T1 = 1 + 2/(2^2) + 3/(3^3) + ... + n/(n^n)$$

$$T2 = 1 + 1/2 + 1/3 + ... + 1/n$$

$$T3 = 1 + 1/(1+2) + 1/(1+2+3) + ... + 1/(1+2+...+n)$$

SV tự giải

17