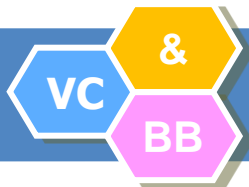


# TIN HỌC CƠ SỞ 2

## CHƯƠNG 4: HÀM VÀ PHẠM VI HOẠT ĐỘNG CỦA BIẾN





# Nội dung

4.1

**Khái niệm**

4.2

**Khai báo, thiết kế, sử dụng hàm**

4.3

**Nguyên tắc hoạt động của hàm**

Lập trình qua 3  
giai đoạn

Lập trình tuyến  
tính

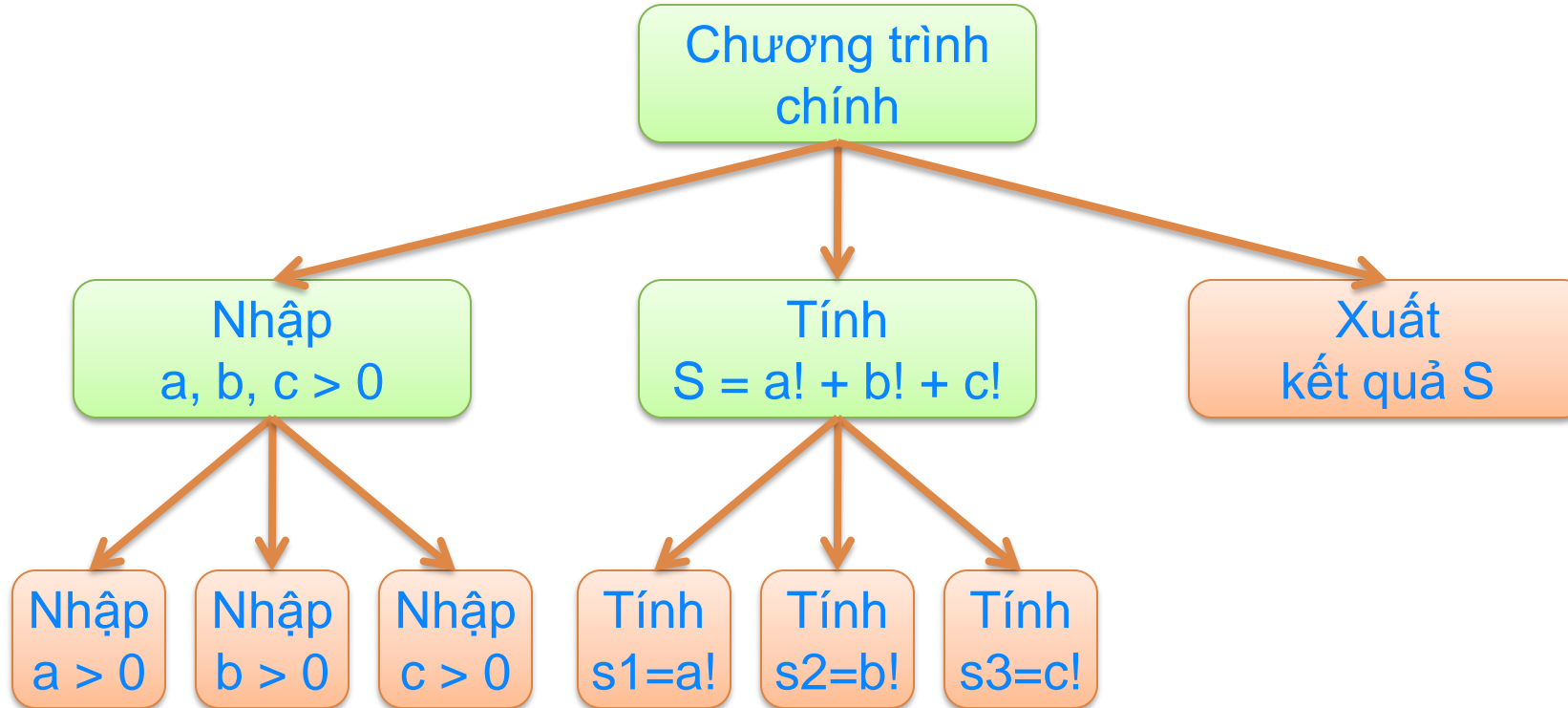
Lập trình thủ  
tục - hàm

Lập trình hướng  
đối tượng

```
vd songuyento.cpp  [*] tinh agiathua.cpp  vdwwhile.cpp  [*] bye.cpp
1  #include<stdio.h>
2  void nhap(int &n)
3  {
4      do {
5          printf("Nhap mot so nguyen duong: ");
6          scanf("%d", &n);
7      } while (n<= 0);
8  }
9  int giaithua(int &a)
10 {
11     int s = 1;
12     for (int i = 2; i <= a ; i++)
13         s = s * i;
14     return s;
15 }
16 main()
17 {
18     int a, b,c, s1,s2,s3,s;
19     nhap(a);
20     nhap(b);
21     nhap(c);
22     s=giaithua(a)+giaithua(b)+giaithua(c);
23     printf("s= %ld",s);
24 }
```

# Đặt vấn đề

- ❖ Ví dụ 1:
- ❖ Viết chương trình tính  $S = a! + b! + c!$  với  $a, b, c$  là 3 số nguyên dương nhập từ bàn phím.



❖ 3 đoạn lệnh nhập  $a, b, c > 0$ 

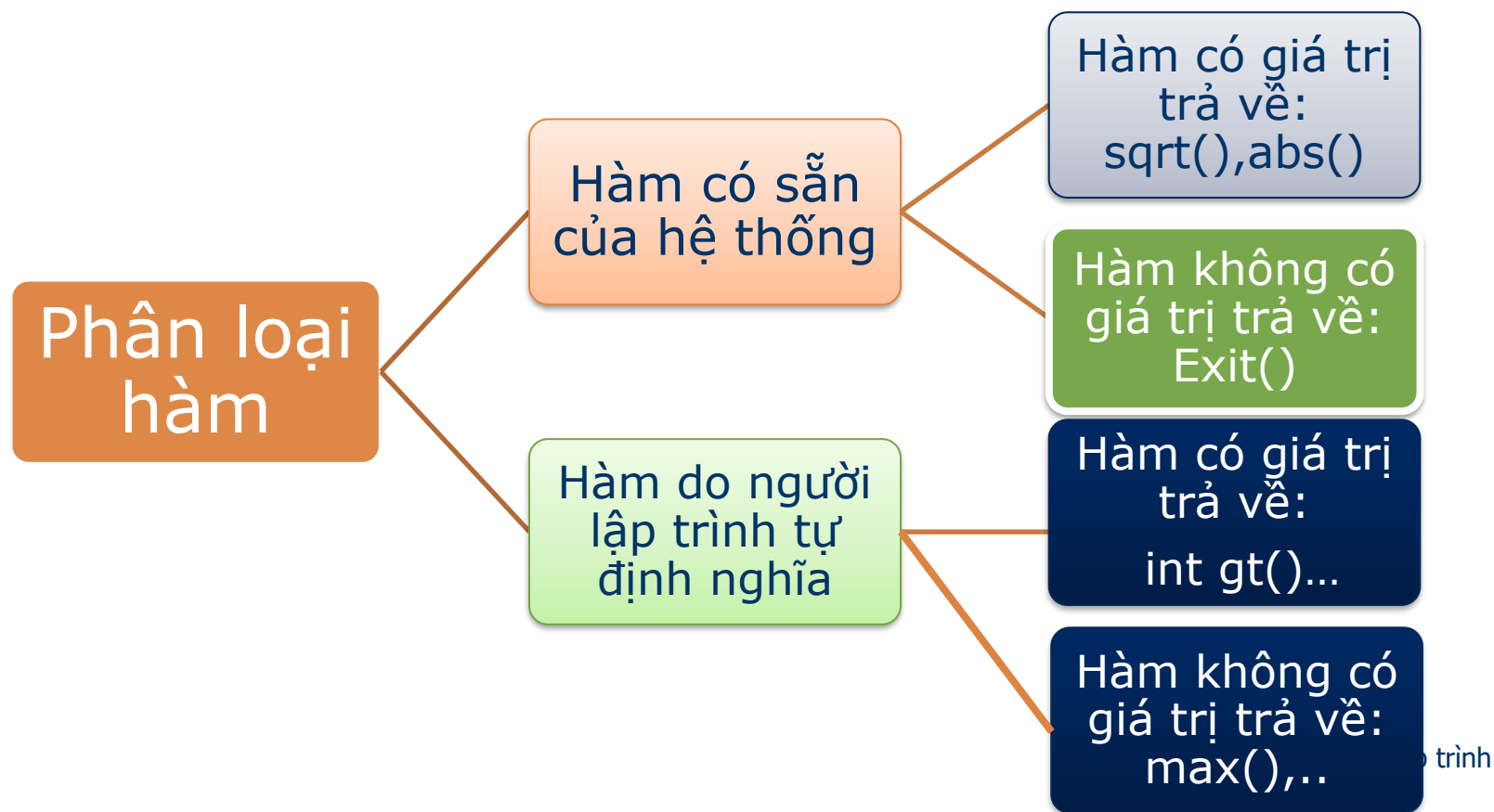
```
do {  
    printf("Nhap mot so nguyen duong: ");  
    scanf("%d", &a);  
} while (a <= 0);  
  
do {  
    printf("Nhap mot so nguyen duong: ");  
    scanf("%d", &b);  
} while (b <= 0);  
  
do {  
    printf("Nhap mot so nguyen duong: ");  
    scanf("%d", &c);  
} while (c <= 0);
```

❖ 3 đoạn lệnh tính  $s1 = a!$ ,  $s2 = b!$ ,  $s3 = c!$

```
{ Tính  $s1 = a! = 1 * 2 * \dots * a$  }  
s1 = 1;  
for (i = 2; i <= a ; i++)  
    s1 = s1 * i;  
  
{ Tính  $s2 = b! = 1 * 2 * \dots * b$  }  
s2 = 1;  
for (i = 2; i <= b ; i++)  
    s2 = s2 * i;  
  
{ Tính  $s3 = c! = 1 * 2 * \dots * c$  }  
s3 = 1;  
for (i = 2; i <= c ; i++)  
    s3 = s3 * i;
```

## 4.1 Khái niệm

Hàm là một nhóm các câu lệnh cùng xử lý một nhiệm vụ nào đó. Còn gọi là chương trình con



# Ý nghĩa và mục đích dùng hàm

```
vd songuyento.cpp  [*] tính agiathua.cpp  vdwwhile.cpp  [*] bye.cpp  
1  #include<stdio.h>  
2  void nhap(int &n)  
3  {  
4      do {  
5          printf("Nhap mot so nguyen duong: ");  
6          scanf("%d", &n);  
7      } while (n<= 0);  
8  }  
9  int giaithua(int &a)  
10 {  
11     int s = 1;  
12     for (int i = 2; i <= a ; i++)  
13         s = s * i;  
14     return s;  
15 }  
16 main()  
17 {  
18     int a, b, c, s1, s2, s3, s;  
19     nhap(a);  
20     nhap(b);  
21     nhap(c);  
22     s=giaithua(a)+giaithua(b)+giaithua(c);  
23     printf("s= %ld", s);
```



## 4.2. Khai báo, thiết kế, sử dụng hàm

### a. Khai báo hàm:

```
<kiểu dữ liệu trả về> <tên hàm> (Danh sách tham số nếu có)
{
    Các câu lệnh thực hiện;
    return giá trị trả về nếu có;
}
```

#### Kiểu dữ liệu trả về

- Hàm không có giá trị trả về: Kiểu dữ liệu là: void (thì không có return)
- Vd  
void tinh tong(int a, int b)  
{ printf("tong = %d", a+b); }

#### Tên hàm

- Theo quy tắc đặt tên
  - Dễ gợi nhớ đến công việc
- Vd: tinh tong, max, tinh\_tong

#### Danh sách tham số

Void tinh tong(int a, int b)  
{ printf("tong = %d", a+b); }

- Nhập tham số: kiểu dữ liệu và tên biến

# Ví dụ 1

## Hàm trả về kết quả

```
#include<stdio.h>
#include<conio.h>
int tinhtong(int a, int b)
{
    return a+b;
}
main()
{
    int x, y,z;
    printf("Nhap x=");scanf("%d",&x);
    printf("Nhap y=");scanf("%d",&y);
    z=tinhtong(x,y);
    printf("tong= %d",z);
    getch();
}
```

## Hàm không trả về kết quả

```
#include<stdio.h>
#include<conio.h>
void tinhtong(int a, int b)
{
    printf("tong= %d",a+b);
}
main()
{
    int x, y;
    printf("Nhap x=");scanf("%d",&x);
    printf("Nhap y=");scanf("%d",&y);
    tinhtong(x,y);
    getch();
}
```

## b. Thiết kế hàm

### Cách 1:

```
#include<stdio.h>
<khai báo nguyên mẫu
hàm>;
main()
{

}
<Định nghĩa hàm>
```

### Cách 2:

```
#include<stdio.h>
<Định nghĩa hàm>
main()
{

}
```

## Ví dụ 2

```
#include<stdio.h>
#include<conio.h>
int tinhtong(int a, int b);
main()
{
    int x, y,z;
    printf("Nhap x=
");scanf("%d",&x);
    printf("Nhap y=
");scanf("%d",&y);
    z=tinhtong(x,y);
    printf("tong= %d",z);
    getch();
}
int tinhtong(int a, int b)
{
    return a+b;
}
```

```
#include<stdio.h>
#include<conio.h>
int tinhtong(int a, int b)
{
    return a+b;
}
main()
{
    int x, y,z;
    printf("Nhap x=
");scanf("%d",&x);
    printf("Nhap y=
");scanf("%d",&y);
    z=tinhtong(x,y);
    printf("tong= %d",z);
    getch();
}
```

## c. Sử dụng hàm

### Hàm có trả về kết quả

- ❖ Gọi trong biểu thức
- ❖ Hoặc kết hợp lệnh in ra màn hình
- ❖ **C=<tên hàm>(danh sách tham số nếu có)**
- ❖ **Printf("%d", <tên hàm>(danh sách tham số nếu có))**
- ❖ **Vd: z=tinhtong(x,y);**
- ❖ **printf("tong = %d",tinhtong(x,y);**

### Hàm không trả về kết quả

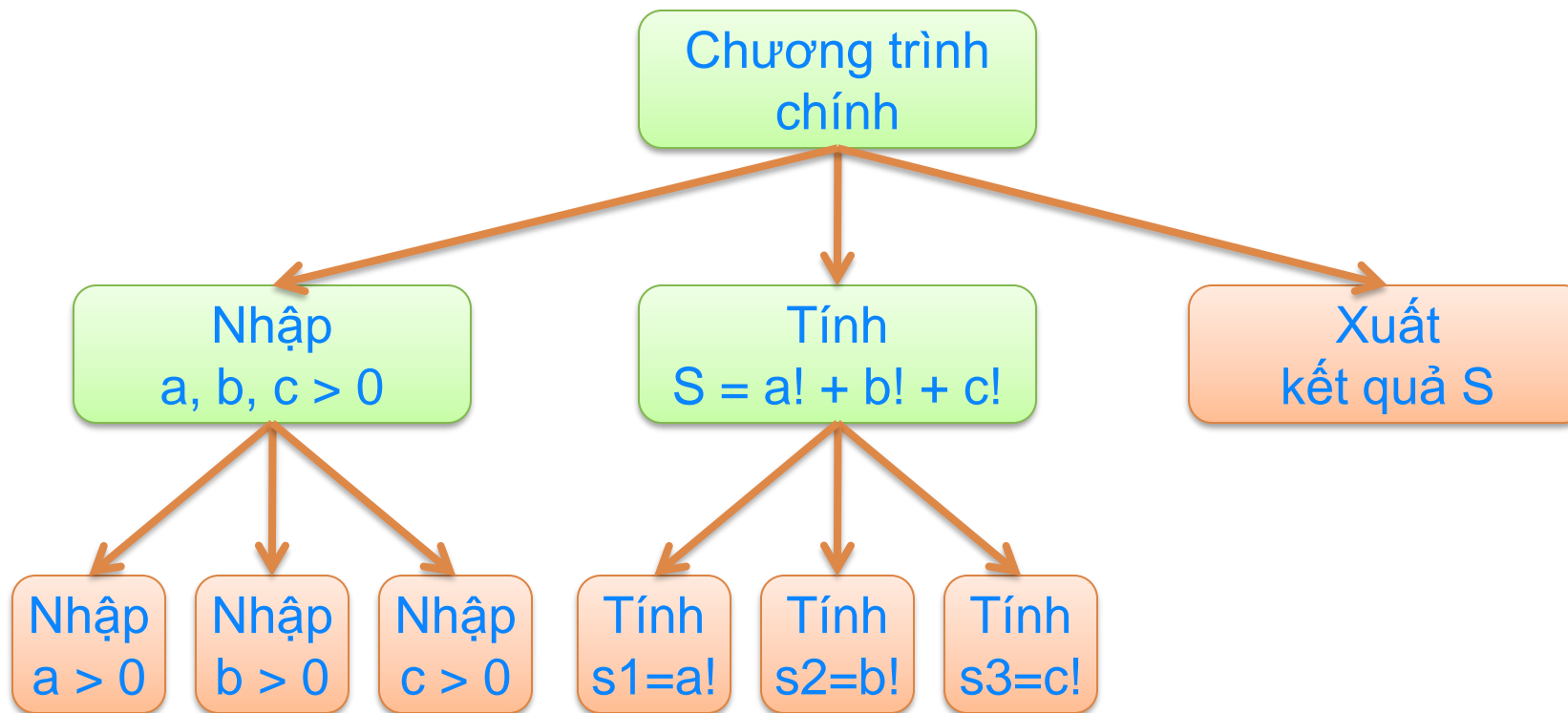
- ❖ Gọi như một lệnh  
**<tên hàm>(danh sách tham số nếu có)**
- Vd:**  
**tinhtong(x,y);**

## Ví dụ 3

```
#include<stdio.h>
#include<conio.h>
int tinhtong(int a, int b);
main()
{
    int x, y,z;
    printf("Nhap x=
");scanf("%d",&x);
    printf("Nhap y=
");scanf("%d",&y);
    z=tinhtong(x,y);
    printf("tong= %d",z);
}
int tinhtong(int a, int b)
{
    return a+b;
}
```

```
#include<stdio.h>
#include<conio.h>
int tinhtong(int a, int b)
{
    return a+b;
}
main()
{
    int x, y,z;
    printf("Nhap x=
");scanf("%d",&x);
    printf("Nhap y=
");scanf("%d",&y);
    z=tinhtong(x,y);
    printf("tong= %d",z);
}
```

- ❖ Viết chương trình tính  $S = a! + b! + c!$  với  $a, b, c$  là 3 số nguyên dương nhập từ bàn phím.



❖ 3 đoạn lệnh nhập  $a, b, c > 0$ 

```
do {  
    printf("Nhap mot so nguyen duong: ");  
    scanf("%d", &a);  
} while (a <= 0);  
  
do {  
    printf("Nhap mot so nguyen duong: ");  
    scanf("%d", &b);  
} while (b <= 0);  
  
do {  
    printf("Nhap mot so nguyen duong: ");  
    scanf("%d", &c);  
} while (c <= 0);
```



❖ 3 đoạn lệnh tính  $s1 = a!$ ,  $s2 = b!$ ,  $s3 = c!$

```
{ Tính  $s1 = a! = 1 * 2 * \dots * a$  }  
s1 = 1;  
for (i = 2; i <= a ; i++)  
    s1 = s1 * i;  
  
{ Tính  $s2 = b! = 1 * 2 * \dots * b$  }  
s2 = 1;  
for (i = 2; i <= b ; i++)  
    s2 = s2 * i;  
  
{ Tính  $s3 = c! = 1 * 2 * \dots * c$  }  
s3 = 1;  
for (i = 2; i <= c ; i++)  
    s3 = s3 * i;
```

## ❖ Giải pháp =&gt; Viết 1 lần và sử dụng nhiều lần

- Đoạn lệnh nhập tổng quát, với  $n = a, b, c$

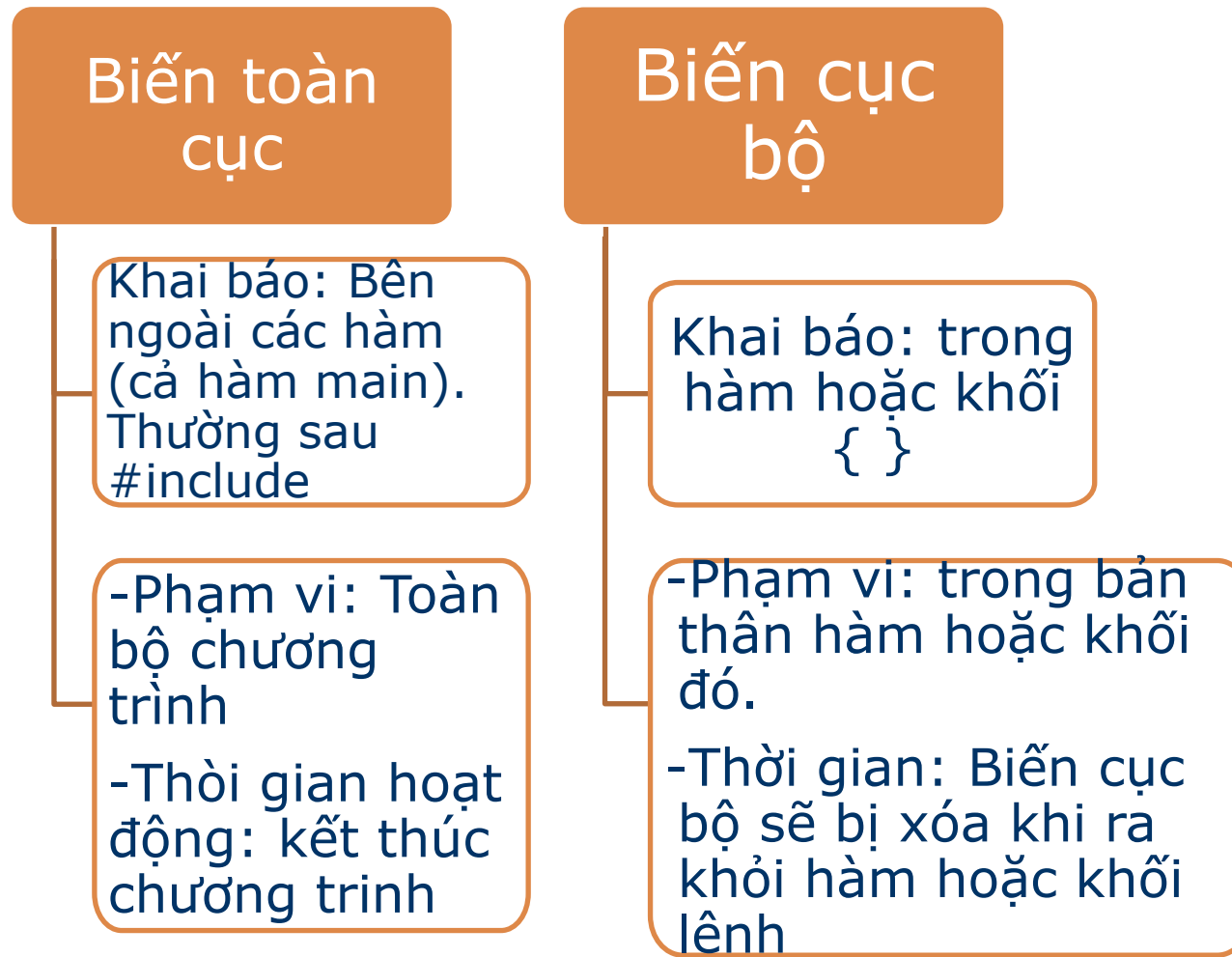
```
do {  
    printf("Nhap mot so nguyen duong: ");  
    scanf("%d", &n);  
} while (n <= 0);
```

- Đoạn lệnh tính giai thừa tổng quát,  $n = a, b, c$

```
{ Tính  $s = n! = 1 * 2 * \dots * n$  }  
s = 1;  
for (i = 2; i <= n ; i++)  
    s = s * i;
```

## 4.3. Nguyên tắc hoạt động của hàm

### a. Phạm vi hoạt động của biến



## Ví dụ 1

```
#include<stdio.h>
#include<conio.h>
int a=6;
void tinh1()
{
    int a=5;
    a=a+10;
    printf("a= %d\n",a);
    if (a>3)
    {
        int a=2;
        a=a+5;
        printf("a= %d\n",a);
    }
}
void tinh2()
{
    a=a+2;
    printf("a= %d\n",a);
}
```

```
main()
{
    int a=3;
    tinh1();
    tinh2();
    printf("a= %d\n",a);
    getch();
}
```



```
a= 15
a= 7
a= 8
a= 3
```

## b. Truyền tham số cho hàm

### Khái niệm tham số

#### Tham số hình thức

- vị trí: được khai báo ở phần đầu của hàm

#### Tham số thực sự

- là các giá trị tương ứng với tham số hình thức trong lời gọi hàm

## b. Truyền tham số cho hàm

### Khái niệm tham số

**Tham số  
hình  
thức**

- Ví dụ: `int  
tinhtong(int a,  
int b)`

**Tham số  
thực sự**

- Ví dụ:  
`tinhtong(x,y)`

## b. Truyền tham số cho hàm

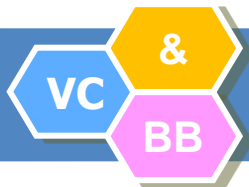
### Khái niệm tham số

Tham số  
hình  
thức

Tham trị:  
void hoanvi  
(int a, int b)

Tham số  
thực sự

Con trỏ:  
hoanvi(int \*a,  
int \*b)



## b. Truyền tham số cho hàm

Truyền tham trị được sử dụng khi không có nhu cầu thay đổi lại tham số

Truyền con trỏ được sử dụng khi có nhu cầu thay đổi lại tham số



## Ví dụ 2: truyền tham trị

```
void hoandoi(int a, int b)
{
    int tg;
    tg=a;
    a=b;
    b=tg;
    printf("trong hoandoi
a= %d va b= %d\n",a, b);
}
```

Tiếp

a,b: gọi là tham số  
hình thức tham trị

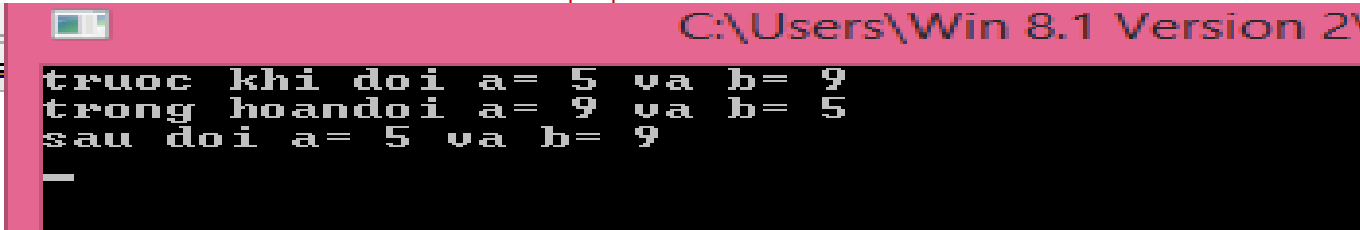
```
main()
{
    int x=5, y=9;
    printf("truc khi doi a= %d
va b= %d\n",x, y);
    hoandoi(x,y);
    printf("sau doi a= %d va
b= %d\n",x, y);
    getch();
}
```

x,y: là tham số  
thực sự truyền  
vào cho tham trị  
a

## Ví dụ 2: truyền tham trị

```
void hoandoi(int a, int b)  
{  
    int tg;  
    tg=a;  
    a=b;  
    b=tg;  
    printf("trong hoandoi  
a= %d va b= %d\n",a, b);  
  
}
```

```
main()  
{  
    int x=5, y=9;  
    printf("truoc khi doi a= %d  
va b= %d\n",x, y);  
    hoandoi(x,y);  
    printf("sau doi a= %d va  
b= %d\n",x, y);  
    getch();  
}
```



```
truoc khi doi a= 5 va b= 9  
trong hoandoi a= 9 va b= 5  
sau doi a= 5 va b= 9
```

```
void hoandoi(int a, int b)
{
    int tg;
    tg=a;
    a=b;
    b=tg;
    printf("trong hoandoi a=
%d va b= %d\n",a, b);
    printf("dia chi cua bien a,b
trong xuly la %p va
%p\n",&a,&b);
}
```

```
main()
{
    int x=5, y=9;
    printf("truoc khi doi a= %d va
b= %d\n",x, y);
    hoandoi(x,y);
    printf("dia chi cua bien x,y trong
xuly la %p va %p\n",&x,&y);
    printf("sau doi a= %d va b=
%d\n",x, y);
    getch();
}
```

```
void hoandoi(int a, int b)
{
    int tg;
    tg=a;
    a=b;
    b=tg; printf("trong hoandoi
a= %d va b= %d\n",a, b);
    printf("dia chi cua bien a b
trong xuly
%p\n",&
```

```
main()
{
    int x=5, y=9;
    printf("truoc khi doi a= %d va
b= %d\n",x, y);
    hoandoi(x,y);
    printf("dia chi cua bien x,y trong
xuly la %p va %p\n",&x,&y);
    printf("sau doi a= %d va b=
```

C:\Users\win 8.1 version 2\Documents\hoanvi.exe

```
truoc khi doi a= 5 va b= 9
trong hoandoi a= 9 va b= 5
dia chi cua bien a,b trong xuly la 000000000023FE20 va 000000000023FE28
dia chi cua bien x,y trong xuly la 000000000023FE4C va 000000000023FE48
sau doi a= 5 va b= 9
-
```

```
void hoandoi(int *a, int *b)
{
    int tg;
    tg=*a;
    *a=*b;
    *b=tg;
    printf("trong hoandoi a= %d va b= %d\n",*a, *b);
    printf("dia chi cua bien x,y a,b trong xuly la %p va %p\n",a,b);
}
```

a,b:gọi là con trỏ

```
main()
{
    int x=5,y=9;
    printf("truoc khi doi a= %d va b= %d\n",x, y);
    hoandoi(&x,&y);
    printf("dia chi cua bien x,y trong xuly la %p va %p\n",&x,&y);
    printf("sau doi a= %d va b= %d\n",x, y);
    getch();
}
```

x,y: là tham số thực sự

## Ví dụ 5: truyền con trỏ

menu

```
void hoandoi(int *a, int *b)
```

```
{
```

```
    int tg;
```

```
    tg=*a;
```

```
    *a=*b;
```

```
main()
```

```
{
```

```
    int x=5,y=9;
```

```
    printf("truc khi doi a= %d va b= %d\n",x, y);
```

```
    hoandoi(&x,&y);
```

C:\Users\Win 8.1 Version 2\Documents\vdkhieu dk 1.exe

```
truc khi doi a= 5 va b= 9
trong hoandoi a= 9 va b= 5
dia chi cua bien x,y a,b trong xuly la 000000000023FE4C va 000000000023FE48
dia chi cua bien x,y trong xuly la 000000000023FE4C va 000000000023FE48
sau doi a= 9 va b= 5
```

y trong

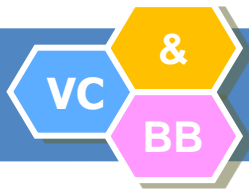
%d va

```
    swap &a, &b;
```

```
}
```

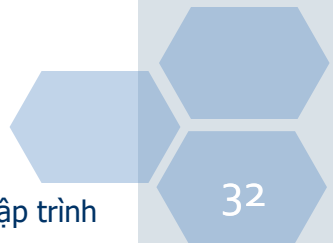
```
#include<stdio.h>
#include<conio.h>
void hoandoi(int &a, int &b)
{
    int tg;
    tg=a;
    a=b;
    b=tg;
    printf("trong hoandoi a= %d va b=
%d\n",a, b);
}
```

```
main()
{
    int x=5, y=9;
    printf("truoc khi doi a=
%d va b= %d\n",x, y);
    hoandoi(x,y);
    printf("sau doi a= %d va
b= %d\n",x, y);
    getch();
}
```



# Tính đệ qui của hàm

- ❖ Một lời gọi hàm được gọi là đệ qui nếu nó gọi đến chính nó.
- ❖ Ví dụ: Nhận xét và cài đặt hàm tính  $n!$  của toán học  
 $n! = 1$  khi  $n=0$ ;  
 $(n-1)! * n$  khi  $n \geq 1$ ;



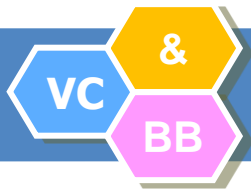


❖ Ví dụ: Nhận xét và cài đặt hàm tính  $n!$  của toán học

$n! = 1$  khi  $n=0$ ;

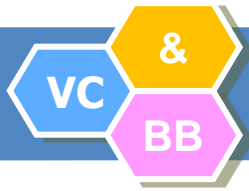
$(n-1)! * n$  khi  $n \geq 1$ ;

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



# Bài tập

- ❖ Viết chương trình nhập vào một số nguyên  $n$  từ bàn phím. Tính và in ra giá trị  $S = 1! + 2! + \dots + n!$



# Bài tập về nhà

- ❖ BTVN ngày 11/3 trên [codeptit.edu.vn](http://codeptit.edu.vn)
- ❖ SV hoàn thành tối thiểu 5/10 bài đạt AC. Không đạt sv sẽ bị trừ 1 điểm chuyên cần
- ❖ Deadline: 12:00 trưa 12/3