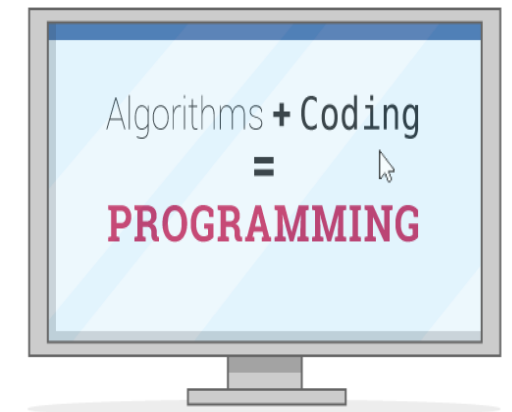
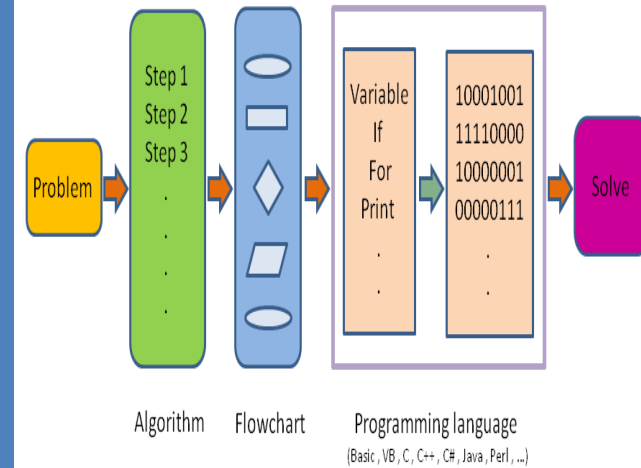


NHẬP MÔN LẬP TRÌNH

Chương 4 HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

GV: Lê Thị Bích Hằng

Trường Đại học Nha Trang



HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

Phạm vi của biến

Truyền tham số cho hàm

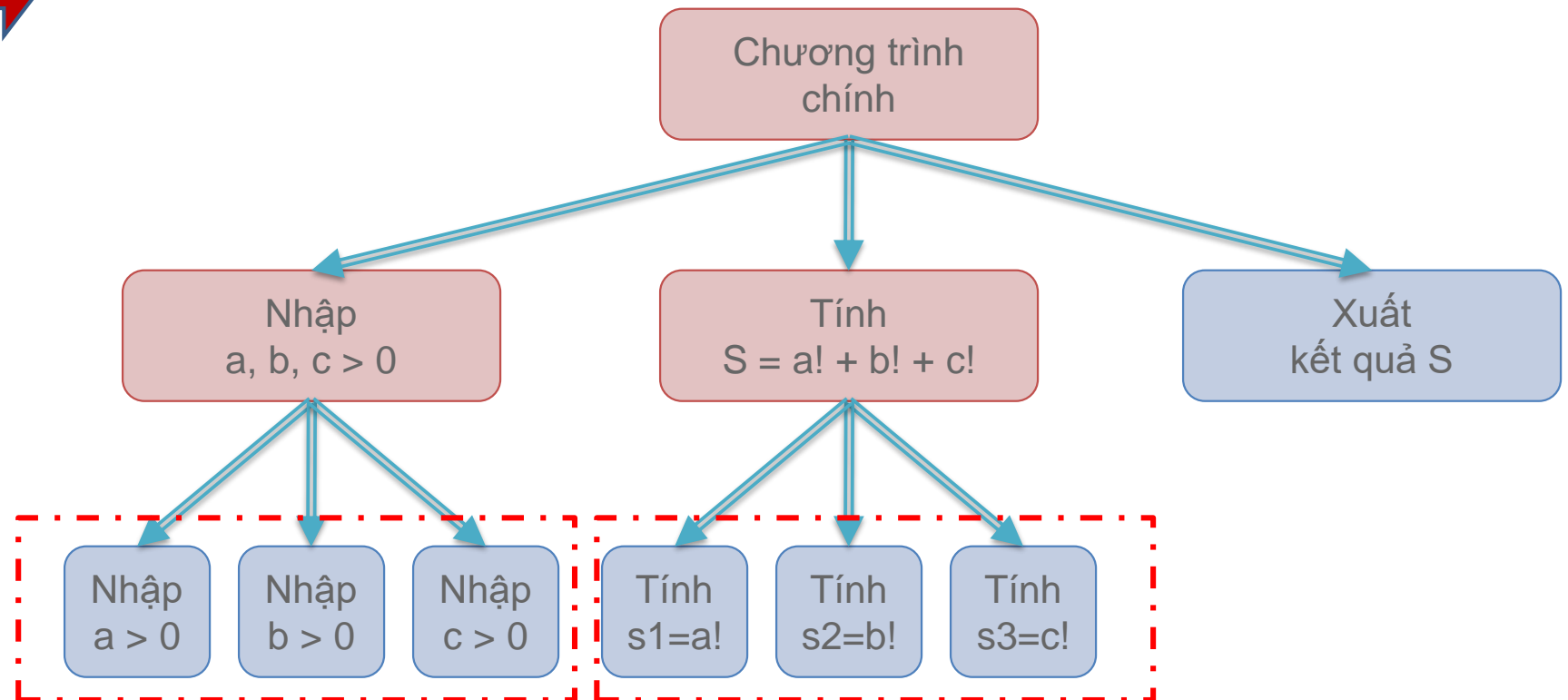
Một số ví dụ minh họa

Tài liệu tham khảo

■ Đặt vấn đề:

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

■ Cách xử lý không phân đoạn:



HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập
trình cấu trúc

Khái niệm và cấu trúc
hàm

Phạm vi của biến

Truyền tham số cho
hàm

Một số ví dụ minh
họa

Tài liệu tham khảo

■ Đặt vấn đề:

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

■ Cách xử lý không phân đoạn:

- 3 đoạn lệnh nhập 3 số $a, b, c > 0$:

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

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

Phạm vi của biến

Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Đặt vấn đề:

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

■ Cách xử lý không phân đoạn:

- 3 đoạn lệnh tính $a!, b!, c!$:

```
// 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;
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo



■ Hạn chế:

- Nhiều đoạn lệnh bị lặp lại
- Tốn thời gian để cập nhật, chỉnh sửa
- Chương trình chính dài dòng, khó hiểu

■ Cách giải quyết:

- Các đoạn lệnh giống nhau được viết một lần
- **Đoạn lệnh nhập tổng quát, với $n = a, b, c$**

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

- **Đoạn lệnh tính $n!$, với $n = a, b, c$**

```
// Tính  $s = n! = 1 * 2 * ... * n$ 
s = 1;
for (i = 2; i <= n ; i++)
    s = s * i;
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập
trình cấu trúc

Khái niệm và cấu trúc
hàm

Phạm vi của biến
Truyền tham số cho
hàm

Một số ví dụ minh
họa

Tài liệu tham khảo



▪ Lập trình cấu trúc:

- Chương trình được phân chia thành nhiều chương trình con (module) có liên kết với nhau.

▪ Phương pháp thiết kế top-down:

- Phương pháp thiết kế top-down tiếp cận bài toán theo hướng từ trên xuống, từ tổng quát đến chi tiết.
- Các bước thực hiện:
 1. Xây dựng sơ đồ khối ở mức đầu tiên bao gồm các module lớn.
 2. Phân rã thuật toán thành các thuật toán con (module).
 3. Xác định chức năng của từng module.
 4. Xác định thứ tự thực hiện các module.
 5. Xác định dữ liệu vào, ra cho từng module.
 6. Xác định cấu trúc của mỗi module.
 7. Viết mã cho từng module.
 8. Liên kết các module (viết mã cho chương trình chính).

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Khái niệm hàm

- Là một đoạn chương trình có **tên**, **đầu vào** và **đầu ra**, có chức năng **giải quyết một số vấn đề chuyên biệt** cho chương trình chính.
- Hàm được gọi nhiều lần với các giá trị tham số khác nhau.
- Ưu điểm: Dễ sửa lỗi, dễ bổ sung, nâng cấp.
- Ví dụ: Hai đoạn lệnh nhập số nguyên dương và tính giai thừa của ví dụ trên được viết lại bằng hàm:

```
void NhapSo(int &n) // hàm nhập số > 0
{
    do {
        printf("Nhap mot so nguyen duong: ");
        scanf("%d", &n);
    } while (n <= 0);
}
```

```
int TinhGiaiThua(int n) //hàm tính giai thừa của một số
{
    int s = 1;
    for (i = 2; i <= n ; i++)
        s = s * i;
    return s;
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Cấu trúc hàm

```
<kiểu trả về> <tên hàm> ([<danh sách tham số hình thức>])  
{  
    <các câu lệnh>  
    [return <biểu thức>;]  
}
```

- <kiểu trả về> : kiểu bất kỳ của C (char, int, long, float,...). Nếu hàm không trả về thì sử dụng khai báo void.
- <tên hàm>: theo quy tắc đặt tên định danh.
- <danh sách tham số> : tham số đầu vào, được khai báo giống khai báo biến, cách nhau bằng dấu phẩy ,
- <biểu thức> : giá trị trả về cho hàm qua lệnh return

```
int TinhGiaiThua(int n)  
{  
    int s = 1;  
    for (i = 2; i <= n ; i++)  
        s = s * i;  
    return s;  
}
```


HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

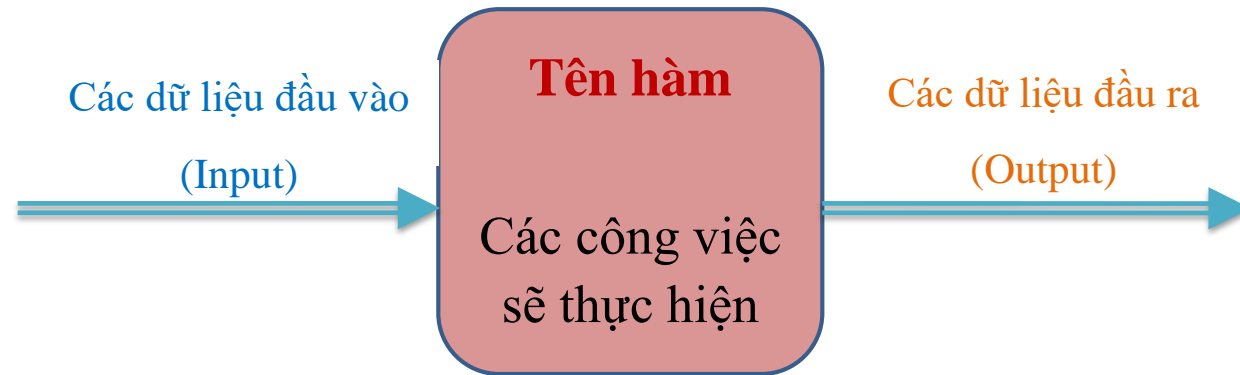
Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

- Quy tắc xây dựng hàm: Cần xác định các thông tin sau:

- Tên hàm
- Công việc mà hàm cần thực hiện
- Các dữ liệu đầu vào (nếu có)
- Các dữ liệu đầu ra (nếu có)



HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

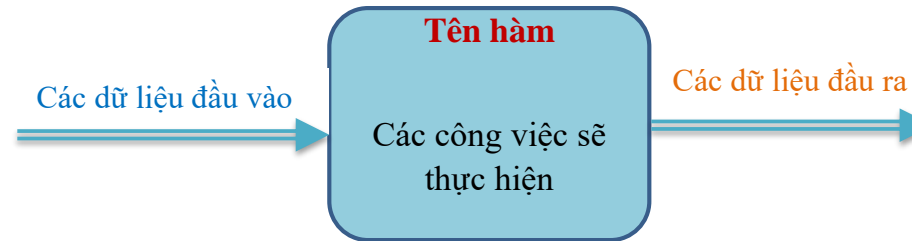
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Quy tắc xây dựng hàm:



• Ví dụ 1: Hàm tính tổng hai số

- **Tên hàm:** `xuatTong`
- **Công việc:** tính và xuất tổng 2 số nguyên
- **Đầu vào:** hai số nguyên `x` và `y`
- **Đầu ra:** không có

```
void xuatTong(int x, int y)
{
    int s;
    s = x + y;
    printf("%d cong %d bang %d", x, y, s);
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

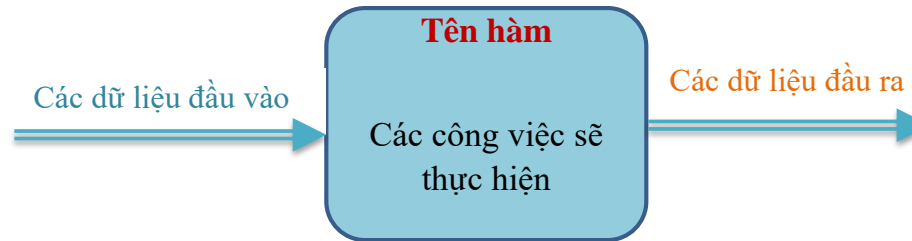
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Quy tắc xây dựng hàm:



• Ví dụ 2: Hàm tính tổng hai số

- **Tên hàm:** tinhTong
- **Công việc:** tính và trả về tổng 2 số nguyên
- **Đầu vào:** hai số nguyên x và y
- **Đầu ra:** một số nguyên có giá trị $x + y$

```
int tinhTong(int x, int y)
{
    int s;
    s = x + y;
    return s;
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

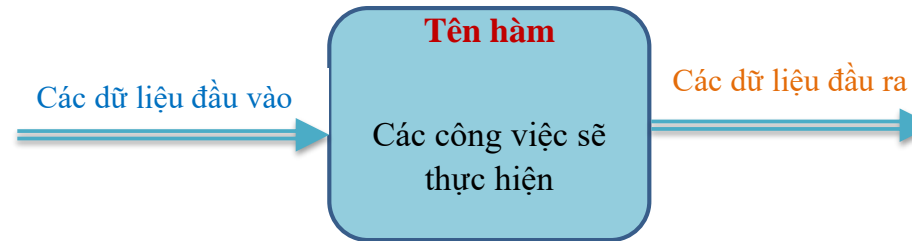
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Quy tắc xây dựng hàm:



• Ví dụ 3: Hàm tính tổng hai số

- **Tên hàm:** nhapXuatTong
- **Công việc:** nhập và xuất tổng 2 số nguyên
- **Đầu vào:** không có
- **Đầu ra:** không có

```

void nhapXuatTong()
{
    int x, y;
    printf("Nhap 2 so nguyen: ");
    scanf("%d%d", &x, &y);
    printf("%d cong %d bang %d", x, y, x + y);
}
  
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm
Một số ví dụ minh họa
Tài liệu tham khảo

Biến toàn cục (*Global variable*)

- Được khai báo bên ngoài tất cả các hàm
- Các biến này có ảnh hưởng đến toàn bộ chương trình
- Chu kỳ sống của biến là bắt đầu chạy chương trình đến lúc kết thúc chương trình

Biến cục bộ (*Local variable*)

- Được khai báo trong hàm hoặc khối { }
- Các biến này chỉ có tác dụng trong bản thân hàm hoặc khối đó (kể cả khối con của nó).
- Biến cục bộ sẽ bị xóa khỏi bộ nhớ khi kết thúc khối khai báo nó

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo



■ Ví dụ:

```
int a;  
int Ham1()  
{  
    int a1;  
}  
  
int Ham2()  
{  
    int a2;  
    {  
        int a21;  
    }  
}  
  
int main()  
{  
    int a3;  
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

Phạm vi của biến

Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Các dạng tham số:

- **Tham số hình thức** (*formal parameter*): là **các biến** được liệt kê trong danh sách tham số (thường nằm tại phần đầu của định nghĩa hàm).
- **Tham số thực** (*actual parameter*): là **các giá trị cụ thể của các tham số hình thức** tại thời gian thực thi chương trình.
 - Tham số thực còn được gọi là đối số (*argument*).

■ Gọi hàm:

Tên_hàm ([Danh_sách_các_tham_số_thực])

- Gọi tên của hàm đồng thời truyền các đối số (hằng, biến, biểu thức) cho các tham số theo đúng thứ tự đã được khai báo trong hàm.
- Các đối số này được cách nhau bằng dấu phẩy ,

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa
Tài liệu tham khảo

■ Gọi hàm:

- Ví dụ:

```
int tinhTong(int x, int y) //khai báo hàm
{
    int s;
    s = x + y;
    return s;
}
```

```
int a, b, c;
b = 3;
c = 5;
a = tinhTong(b, c); //gọi hàm
```

Hàm `tinhTong` nhận giá trị của `s`

- Các dạng gọi hàm khác:

```
int a, b = 3, c = 5;
a = tinhTong(3, 5);
a = tinhTong(b, 5);
```


HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc

Khái niệm và cấu trúc hàm

Phạm vi của biến

Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo



Các cách truyền tham số

Truyền theo giá trị
(*Call by value*)

Truyền theo địa chỉ
(*Call by address*)

Truyền theo tham chiếu
(*Call by reference*)

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa
Tài liệu tham khảo



Truyền theo giá trị

- Truyền đối số cho hàm ở dạng giá trị.
- Có thể truyền hằng/biến/biểu thức nhưng hàm chỉ nhận giá trị.
- Được sử dụng khi không có nhu cầu thay đổi giá trị của tham số sau khi thực hiện hàm.
- Ví dụ:

```
void truyenGiaTri (int a)
{
    a++;
}

int main()
{
    int x = 5;
    truyenGiaTri (x);
    printf("%d", x);    // In ra: 5
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa
Tài liệu tham khảo



Truyền theo địa chỉ

- Truyền đối số cho hàm ở dạng địa chỉ (con trỏ).
- Không được truyền giá trị cho tham số này.
- Được sử dụng khi có nhu cầu thay đổi giá trị của tham số sau khi thực hiện hàm.
- Ví dụ:

```
void truyenDiaChi (int *a)
{
    *a++;
}

int main()
{
    int x = 5;
    truyenDiaChi (&x);
    printf("%d", x);           // In ra: 6
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa
Tài liệu tham khảo



▪ Truyền theo tham chiếu

- Còn được gọi là **Truyền theo tham biến**.
- Truyền đối số cho hàm **ở dạng địa chỉ** (con trỏ). Được bắt đầu bằng **&** trong khai báo.
- **Không được truyền giá trị** cho tham số này.
- Được sử dụng khi **có nhu cầu thay đổi giá trị của tham số** sau khi thực hiện hàm.
- Ví dụ:

```
void truyenThamChieu (int &a)
{
    a++;
}
```

```
int main()
{
    int x=5;
    truyenThamChieu (x);
    printf("%d", x); } // In ra: 6
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập
trình cấu trúc

Khái niệm và cấu trúc
hàm

Phạm vi của biến

Truyền tham số cho
hàm

Một số ví dụ minh
họa

Tài liệu tham khảo

■ Ví dụ 1:

- Viết hàm tính tổng các chữ số của một số nguyên > 0 .
Viết chương trình nhập vào một số nguyên > 0 , rồi dùng hàm trên kiểm tra xem số đó có chia hết cho 3 không?

```
#include<stdio.h>
int tongChuSo (int n)  // n là tham trị
{
    int soDV, tong = 0;
    while(n > 0)
    {
        soDV = n %10;
        tong = tong + soDV;
        n = n /10;
    }
    return tong;
}
int main()
{
    int n;
    printf("Nhap vao mot so n:");
    scanf("%d",&n);
    if(tongChuSo(n) % 3 == 0)
        printf("So %d chia het cho 3",n);
    else
        printf("So %d khong chia het cho 3",n);
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Ví dụ 2:

- Viết chương trình có sử dụng hàm, in và tính tổng các số chính phương nhỏ hơn n ($5 < n \leq 100$).

```
#include<stdio.h>
#include<math.h>
void nhapSo(int &n)    // n là tham chiếu
{
    do{
        printf("Nhap vao n (0<n<=100):");
        scanf("%d",&n);
    }while(!(n>5 && n<=100));
}
int kt_SoCP (int n)    //n là tham trị
{
    int socan = pow(n, 0.5);
    if (socan*socan == n)
        return 1;
    else
        return 0;
}
////////////////////////////////////
int main()
{
    int i, n;
    nhapSo(n);
    printf("\nCac so chinh phuong nho hon %d la:",n);
    for(i = 1; i < n; i++)
        if(kt_SoCP(i))
            printf("%5d",i);
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập trình cấu trúc
Khái niệm và cấu trúc hàm

Phạm vi của biến
Truyền tham số cho hàm

Một số ví dụ minh họa

Tài liệu tham khảo

■ Ví dụ 3: (Sử dụng prototype)

- Viết chương trình có sử dụng hàm, in và tính tổng các số chính phương nhỏ hơn n ($5 < n \leq 100$).

```
#include<stdio.h>
#include<math.h>
void nhapSo(int &n);    // n là tham chiếu
int kt_SoCP (int n);    //n là tham trị
////////////////////////////////////
int main()
{
    int i, n;
    nhapSo(n);
    printf("\nCac so chinh phuong nho hon %d la:",n);
    for(i = 1; i < n; i++)
        if(kt_SoCP(i))
            printf("%5d",i);
}
////////////////////////////////////
void nhapSo(int &n)    // n là tham chiếu
{
    do{ printf("Nhap vao n (0<n<=100):");
        scanf("%d",&n);
    }while(!(n>5 && n<=100));
}
int kt_SoCP (int n)    //n là tham trị
{
    int socan = pow(n, 0.5);
    if (socan*socan == n) return 1;
    else return 0;
}
```

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Phương pháp lập
trình cấu trúc

Khái niệm và cấu trúc
hàm

Phạm vi của biến

Truyền tham số cho
hàm

Một số ví dụ minh
họa

Tài liệu tham khảo



- [1] Quách Tuấn Ngọc, *Ngôn ngữ lập trình C*, NXB Giáo dục, 2003
- [2] Trần Đan Thư, *Giáo trình lập trình C tập 1&2*, NXB ĐHQG TP. Hồ Chí Minh
- [3] Trần Đan Thư, *Nhập môn lập trình*, NXB KH&KT, 2019
- [4] Phạm Văn Ất, *Kỹ thuật lập trình C căn bản và nâng cao*, NXB Hồng Đức 2009
- [5] B. W. Kernighan and D. M. Ritchie, *C Programming Language, 2nd edition*, Prentice Hall, 1998
- [6] Lê Thị Bích Hằng, *Tài liệu tham khảo Nhập môn lập trình*, Lưu hành nội bộ
- [7] Đặng Bình Phương, *Bài giảng Nhập môn lập trình*