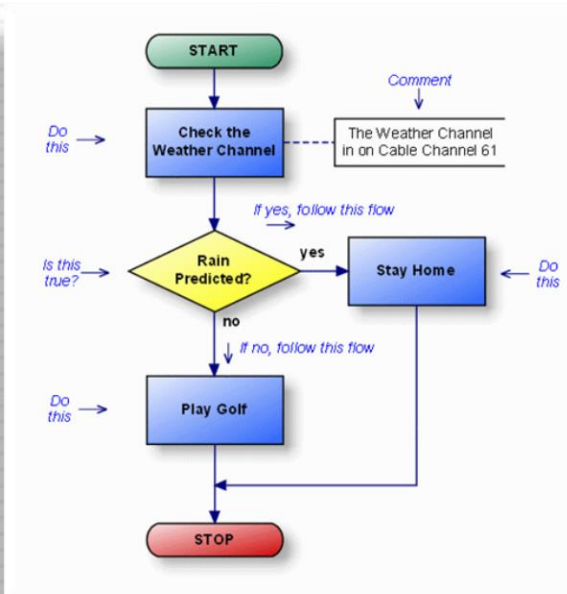


NHẬP MÔN LẬP TRÌNH

CHƯƠNG 4: HÀM (FUNCTION)



GV: Phạm Nguyễn Sơn Tùng

Email: pnstung@fit.hcmus.edu.vn

TẠI SAO PHẢI SỬ DỤNG HÀM?

Ví dụ 1: Viết chương trình nhập vào số n, kiểm tra n có phải là số nguyên tố hay không?

Nếu n không phải là số nguyên tố thì chuyển n về số nguyên tố gần nhất.

```
#include <iostream>
using namespace std;

int main()
{
    while (true) //DEMO
    {
        int dem = 0;
        for (int i = 1; i <= a; i++)
        {
            if (a % i == 0)
                dem++;
        }
        if (dem == 2)
            break;
        a++;
    }
    return 0;
}
```

CÚ PHÁP VÀ CÁCH SỬ DỤNG

Hàm sẽ giúp chia nhỏ chương trình, giúp trả về các kết quả cần thiết để giúp hàm main (hàm chính) giải quyết yêu cầu của bài toán.

Ưu điểm:

- Giải quyết nhiều công việc giống nhau lặp đi lặp lại.
- Giúp chương trình rõ ràng và tường minh hơn.

Khuyết điểm:

- Nhiều quy tắc cần phải tuân thủ.
- Chương trình phức tạp hơn nếu việc chia nhỏ bài toán không phù hợp.

CÚ PHÁP VÀ CÁCH SỬ DỤNG

```
#include <...> //Khai báo thư viện
using namespace std;

//Các hàm con được viết tại đây

int main() //Hàm chính
{
    //Các câu lệnh viết tại đây
}
```

```
<kiểu trả về> <tên hàm>(<danh sách tham số đầu vào>)
{
    //các câu lệnh cần thực hiện
    return <giá trị trả về phù hợp>;
}
```

TẠI SAO PHẢI SỬ DỤNG HÀM?

Bài 1: Viết chương trình nhập vào số n , kiểm tra n có phải là số nguyên tố hay không?

Nếu n không phải là số nguyên tố thì chuyển n về số nguyên tố gần nhất.

```
#include <iostream>
using namespace std;

bool KiemTraNguyenTo(int n)
{
    int dem = 0;
    for (int i = 1; i <= n; i++)
        if (n % i == 0)
            dem++;
    if (dem == 2)
        return true;
    return false;
}
```

TẠI SAO PHẢI SỬ DỤNG HÀM?

```
int main()
{
    int n = 0;
    cout << "Nhap gia tri n: ";
    cin >> n;
    int a = n;
    int b = n;
    while (true)
    {
        if (KiemTraNguyenTo(a) == true)
            break;
        a++;
    }
    while (true)
    {
        if (KiemTraNguyenTo(b) == true)
            break;
        b--;
        if (b < 0)
            break;
    }
    if ((a - n) > (n - b))
        cout << "So can tim la: " << b;
    else
        cout << "So can tim la: " << a;
    return 0;
}
```

CÚ PHÁP VÀ CÁCH SỬ DỤNG

```
<kiểu trả về> <tên hàm>(<danh sách tham số đầu vào>)  
{  
    //các câu lệnh cần thực hiện  
    return <giá trị trả về phù hợp>;  
}
```

Kiểu trả về:

- **void:** không có giá trị trả về, có thể xuất kết quả bên trong hàm.
- **bool / int / float / char / string** trả về giá trị kết quả có kiểu dữ liệu tương ứng với bài toán (chỉ trả về được 1 giá trị theo kiểu dữ liệu)

CÚ PHÁP VÀ CÁCH SỬ DỤNG

```
<kiểu trả về> <tên hàm>(<danh sách tham số đầu vào>)  
{  
    //các câu lệnh cần thực hiện  
    return <giá trị trả về phù hợp>;  
}
```

Tên hàm:

- **Cách đặt tên:** Đặt tên theo qui ước sao cho phản ánh đúng chức năng thực hiện của hàm. Tránh đặt tên vắn tắt không rõ nghĩa.
- **Ví dụ:** TinhTong(), KiemTraNguyenTo(int n)...

CÚ PHÁP VÀ CÁCH SỬ DỤNG

```
<kiểu trả về> <tên hàm>(<danh sách tham số đầu vào>)  
{  
    //các câu lệnh cần thực hiện  
    return <giá trị trả về phù hợp>;  
}
```

Danh sách tham số đầu vào:

- **Không có tham số đầu vào:** Hàm không nhất thiết phải có tham số đầu vào, hàm có thể không có vấn xử lý được yêu cầu đề bài.
- **Có tham số đầu vào:** Hàm có thể có nhiều tham số đầu vào, mỗi tham số đầu vào cách nhau bởi dấu phẩy.

CÚ PHÁP VÀ CÁCH SỬ DỤNG

```
<kiểu trả về> <tên hàm>(<danh sách tham số đầu vào>)  
{  
    //các câu lệnh cần thực hiện  
    return <giá trị trả về phù hợp>;  
}
```

Danh sách tham số đầu vào:

- **Truyền tham trị (Call by Value):** Truyền giá trị để giải quyết yêu cầu của bài toán.
- **Truyền tham chiếu (Call by Reference):** Dùng để thay đổi giá trị của tham số sau khi thực hiện hàm, cũng có thể dùng cách này để trả về nhiều giá trị cùng lúc.

CÚ PHÁP VÀ CÁCH SỬ DỤNG

Ví dụ 2: Người dùng nhập vào biến n . Kiểm tra xem từ 1 đến n tồn tại bao nhiêu số nguyên tố.

Cách 1: Xuất kết quả ngay bên trong hàm. Hàm có tham số đầu vào.

Cách 2: Xuất kết quả trong hàm main, không có tham số đầu vào.

MỘT SỐ LƯU Ý KHI SỬ DỤNG HÀM

Biến toàn cục: khai báo bên ngoài tất cả các hàm, kể cả hàm main và có tác dụng lên toàn bộ chương trình.

Biến cục bộ: khai báo bên trong hàm hoặc bên trong khối lệnh { } và chỉ có tác dụng trong bản thân hàm hoặc khối lệnh đó. Biến cục bộ sẽ bị xóa khỏi bộ nhớ khi kết thúc khối khai báo nó.

BÀI TẬP ỨNG DỤNG

Bài 1: Viết chương trình tìm USCLN của 2 số a và b (a, b nhập vào từ bàn phím).

Bài 2: Viết chương trình nhập 2 số chính và in số chính phương nằm trong đoạn 2 số đã nhập vào.

Bài 3: Nhập vào 1 phân số, hãy tối giản phân số đó.

Bài 4: Viết chương trình in ra các số hoàn thiện nhỏ hơn n .

Bài 5: Viết hàm nhập vào số n , tính hiệu giữa số n và số nghịch đảo của n .

Bài 6: Viết hàm nhập vào số n , kiểm tra n có phải là số đối xứng hay không.