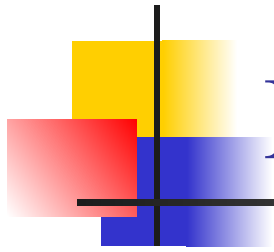


Khái niệm cơ bản C

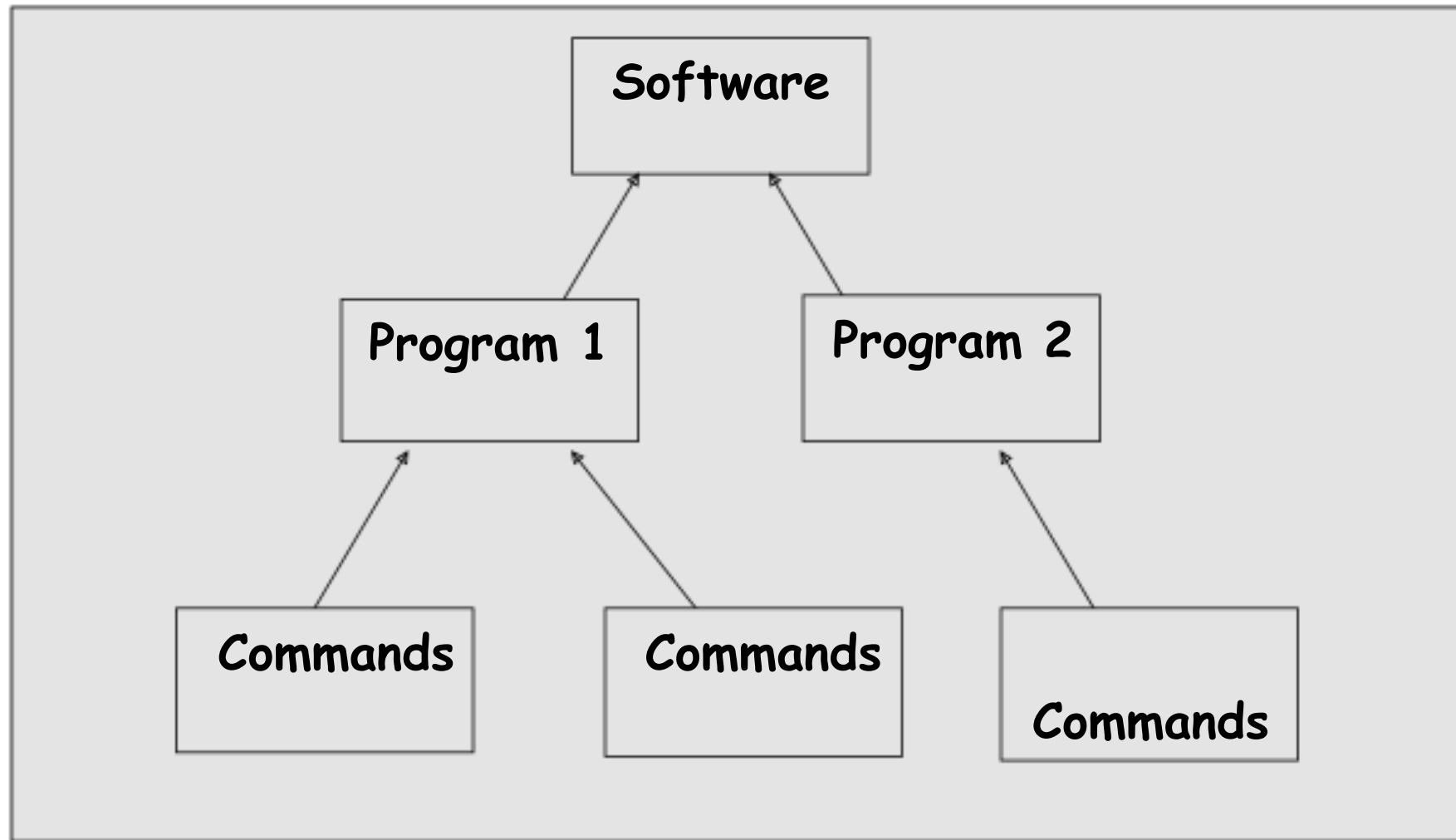
Chương 1

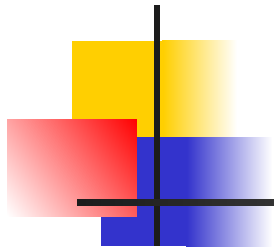


Mục Tiêu

- Phân biệt sự khác nhau giữa Câu lệnh, Chương trình và Phần mềm
- Biết được quá trình hình thành ngôn ngữ C
- Biết được khi nào dùng C và tại sao
- Nắm được cấu trúc ngôn ngữ C
- Hiểu rõ khái niệm giải thuật (algorithms)
- Vẽ lưu đồ (flowchart)
- Sử dụng được các ký hiệu dùng trong lưu đồ

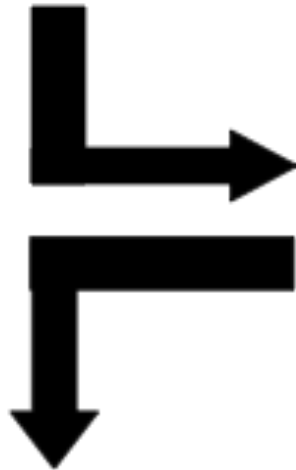
Phần mềm, chương trình, câu lệnh





Bắt đầu C

BPCL – Martin Richards

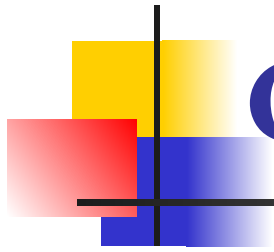


B – Ken Thompson

C – Dennis Ritchie

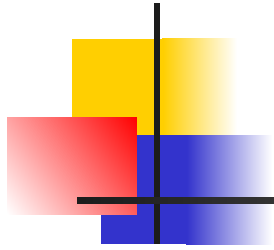


Bell Laboratories, Inc.



Các lĩnh vực ứng dụng của C

- C được dùng để lập trình hệ thống
- Một chương trình hệ thống làm thành một phần hệ điều hành hoặc các tiện ích hỗ trợ của hệ điều hành
- Hệ điều hành (Operating Systems), trình thông dịch (Interpreters), trình soạn thảo (Editors), trình Hợp Ngữ (Assembly) được gọi là chương trình hệ thống
- Hệ điều hành UNIX được phát triển dựa vào C
- Có các trình biên dịch dành cho hầu hết các loại hệ thống PC

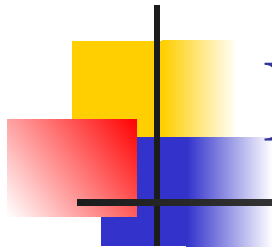


Ngôn ngữ cấp trung

Ngôn ngữ cấp
cao

C

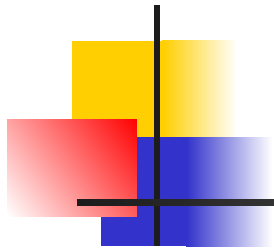
Ngôn ngữ hợp
ngữ



Ngôn ngữ có cấu trúc

- C cho phép tổng hợp mã lệnh và dữ liệu
- Nó có khả năng tập hợp và ẩn đi tất cả thông tin, lệnh khởi phần còn lại của chương trình để dùng cho những tác vụ riêng
- Chương trình C có thể được chia nhỏ thành những hàm (functions) hay những khối mã (code blocks).

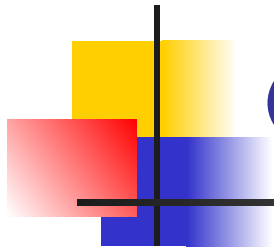
```
do  
{  
    i = i + 1;  
    .  
    .  
    .  
} while (i < 40);
```



Đặc điểm của C

- C có 32 từ khóa
 - Những từ khóa này kết hợp với cú pháp của C hình thành ngôn ngữ C
 - Các quy tắc được áp dụng cho các chương trình
- C Tất cả từ khóa là chữ thường
 - Đoạn mã trong chương trình C có phân biệt chữ thường, chữ hoa, do while khác DO WHILE
 - Từ khóa không thể dùng đặt tên biến (variable name) hoặc tên hàm (function name)

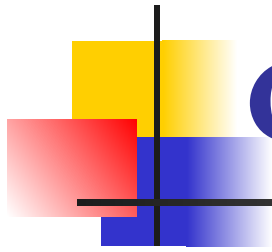
```
main()  
{  
/* This is a sample Program*/  
    int i,j;  
    i=100;  
    j=200;  
    :  
}
```

Cấu trúc chương trình C

main()

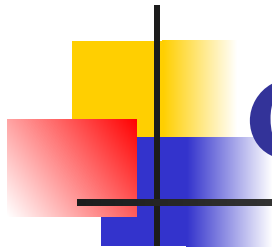
- Chương trình C được chia nhỏ thành những đơn vị gọi là hàm
- Không kể có bao nhiêu hàm trong chương trình, Hệ điều hành luôn trao quyền điều khiển cho hàm `main()` khi một chương trình C được thực thi.
- Theo sau tên hàm là dấu ngoặc đơn
- Dấu ngoặc đơn có thể có chứa hay không chứa những tham số



Cấu trúc chương trình C (tt.)

Dấu phân cách {...}

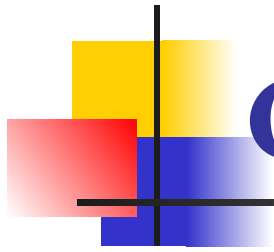
- Sau phần đầu hàm là dấu ngoặc xoắn mở {
- Nó cho biết việc thi hành lệnh trong hàm bắt đầu
- Tương tự, dấu ngoặc xoắn đóng } sau câu lệnh cuối cùng trong hàm chỉ ra điểm kết thúc của hàm



Cấu trúc chương trình C (tt.)

Dấu kết thúc câu lệnh ... ;

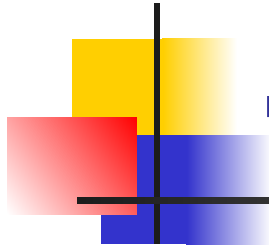
- Một câu lệnh trong C được kết thúc bằng dấu chấm phẩy ;
- Trình biên dịch C không hiểu việc xuống dòng, khoảng trắng hay tab
- Một câu lệnh không kết thúc bằng dấu chấm phẩy sẽ được xem như dòng lệnh lỗi trong C



Cấu trúc chương trình C (tt.)

/*Dòng chú thích*/

- Những chú thích thường được viết để mô tả công việc của một lệnh đặc biệt, một hàm hay toàn bộ chương trình
- Trình biên dịch sẽ bỏ qua phần chú thích
- Trong trường hợp chú thích nhiều dòng, nó sẽ bắt đầu bằng ký hiệu **/*** và kết thúc là ***/**

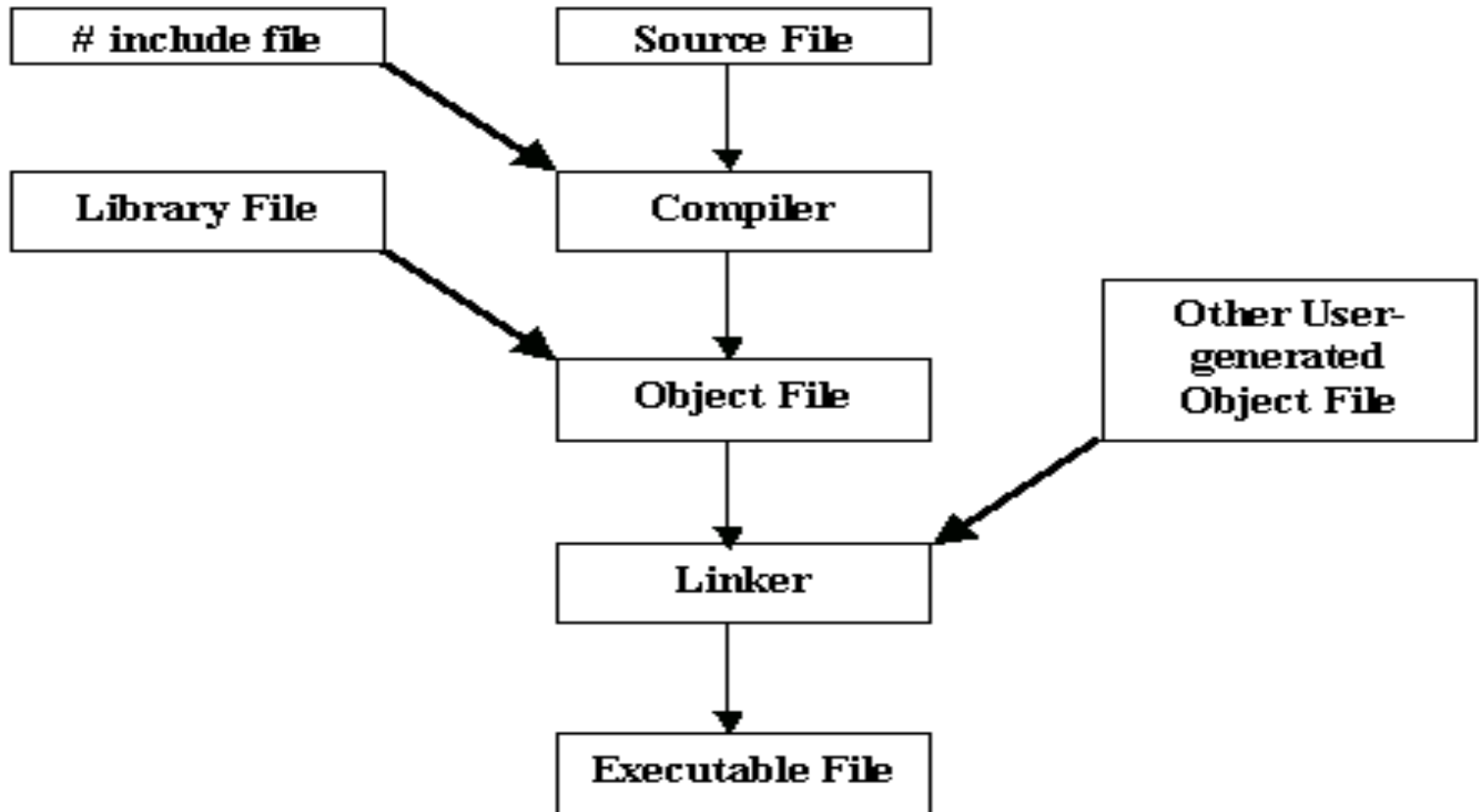


Thư viện C



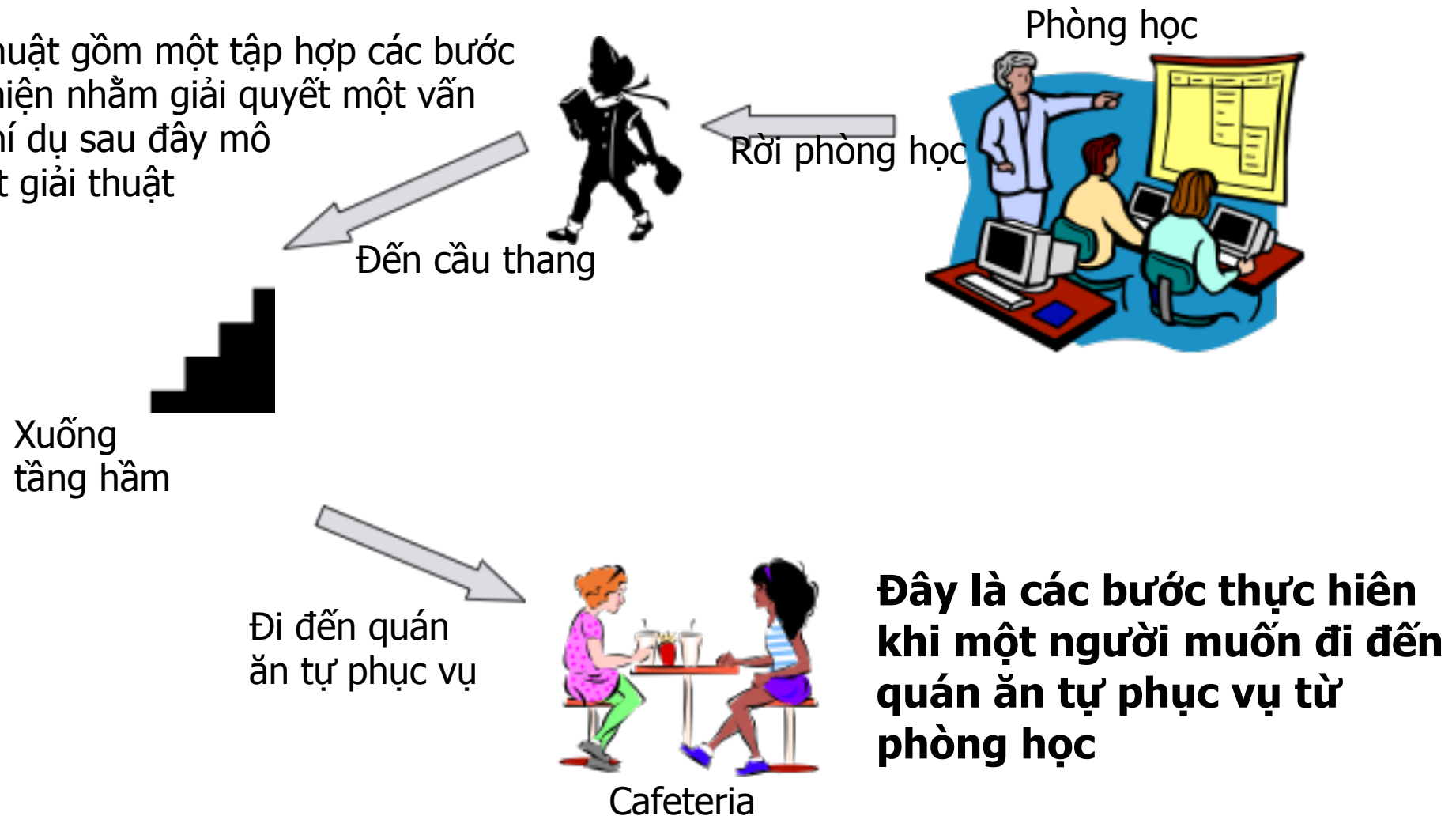
- Tất cả trình biên dịch C đều chứa một thư viện hàm chuẩn
- Một hàm được viết bởi lập trình viên có thể được đặt trong thư viện và được dùng khi cần thiết
- Một số trình biên dịch cho phép thêm hàm vào thư viện chuẩn
- Một số trình biên dịch yêu cầu tạo một thư viện riêng

Biên dịch và thi hành chương trình



Các bước lập trình giải quyết vấn đề

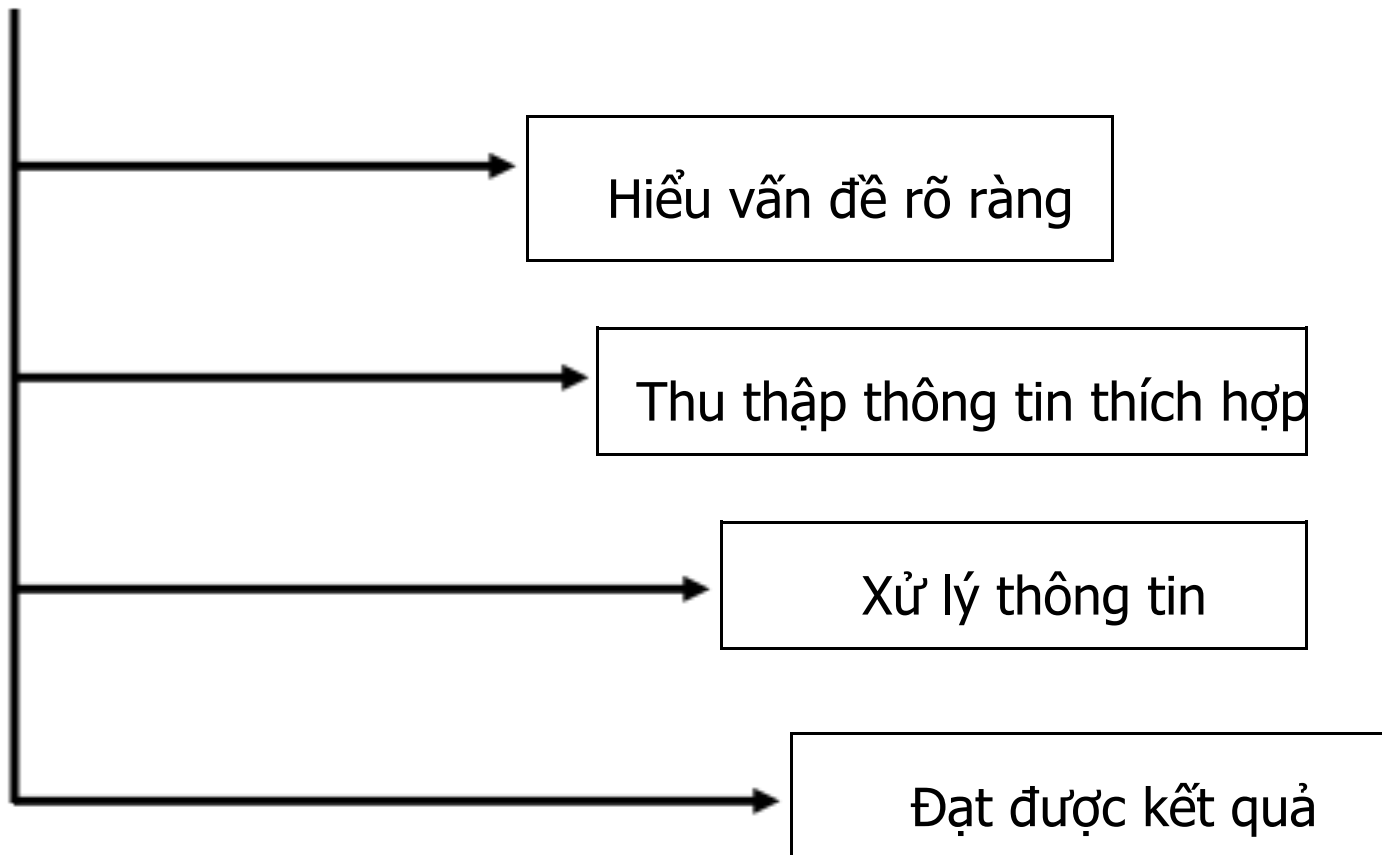
Giải thuật gồm một tập hợp các bước thực hiện nhằm giải quyết một vấn đề. Thí dụ sau đây mô tả một giải thuật





Giải quyết một vấn đề

Để giải quyết một vấn đề





Mã giả (Pseudocode)

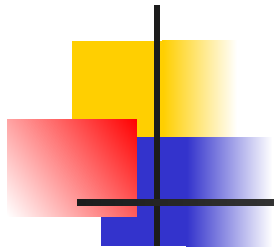
Không là mã thật. Một phương pháp viết giải thuật sử dụng một tập hợp các từ tương tự mã thật

BEGIN

DISPLAY 'Hello World !'

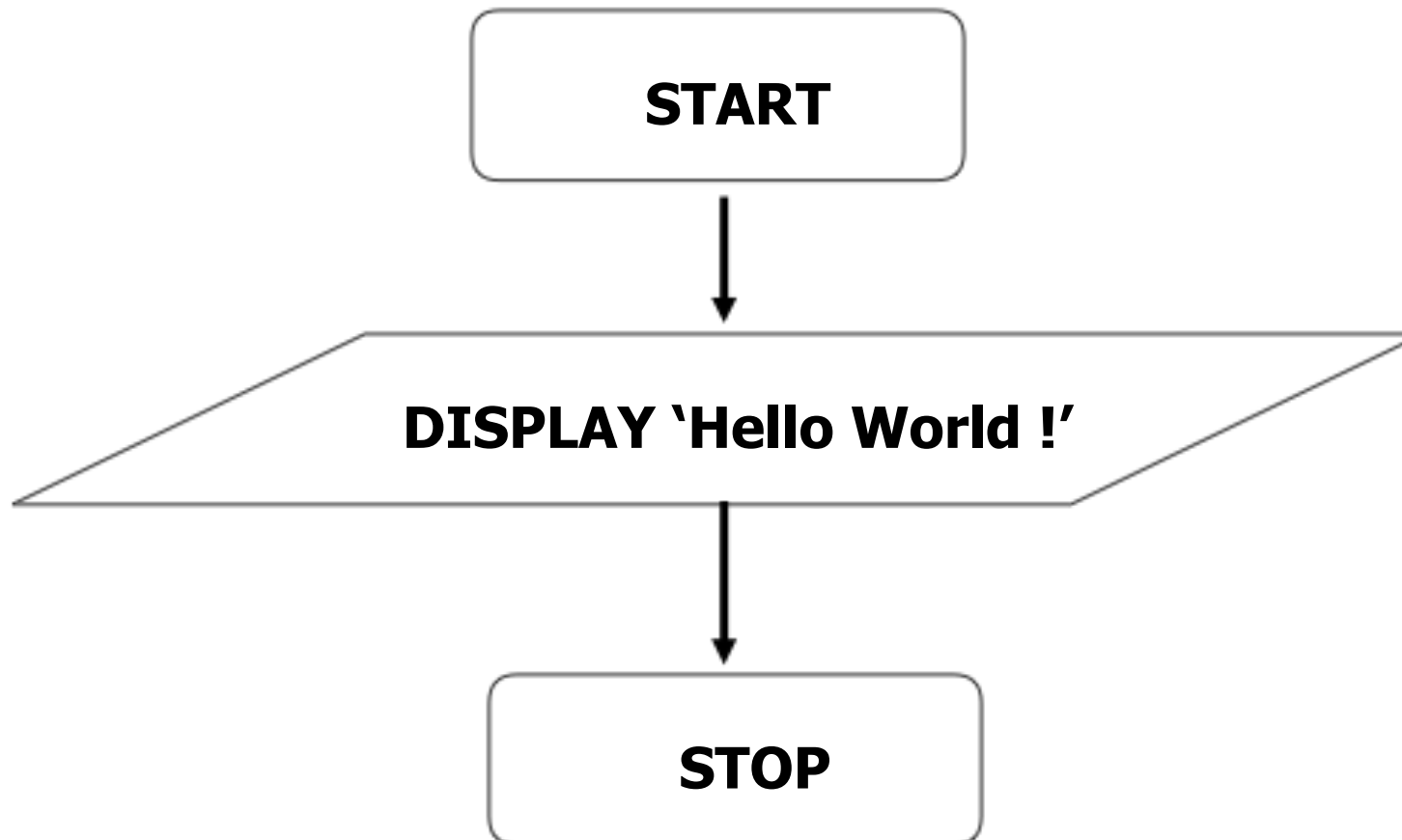
END

Mỗi đoạn mã giả phải bắt đầu với một từ BEGIN
Để hiển thị giá trị nào đó, từ DISPLAY được dùng
Mã giả kết thúc với từ END


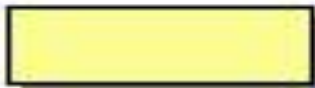






Lưu đồ (Flowcharts)

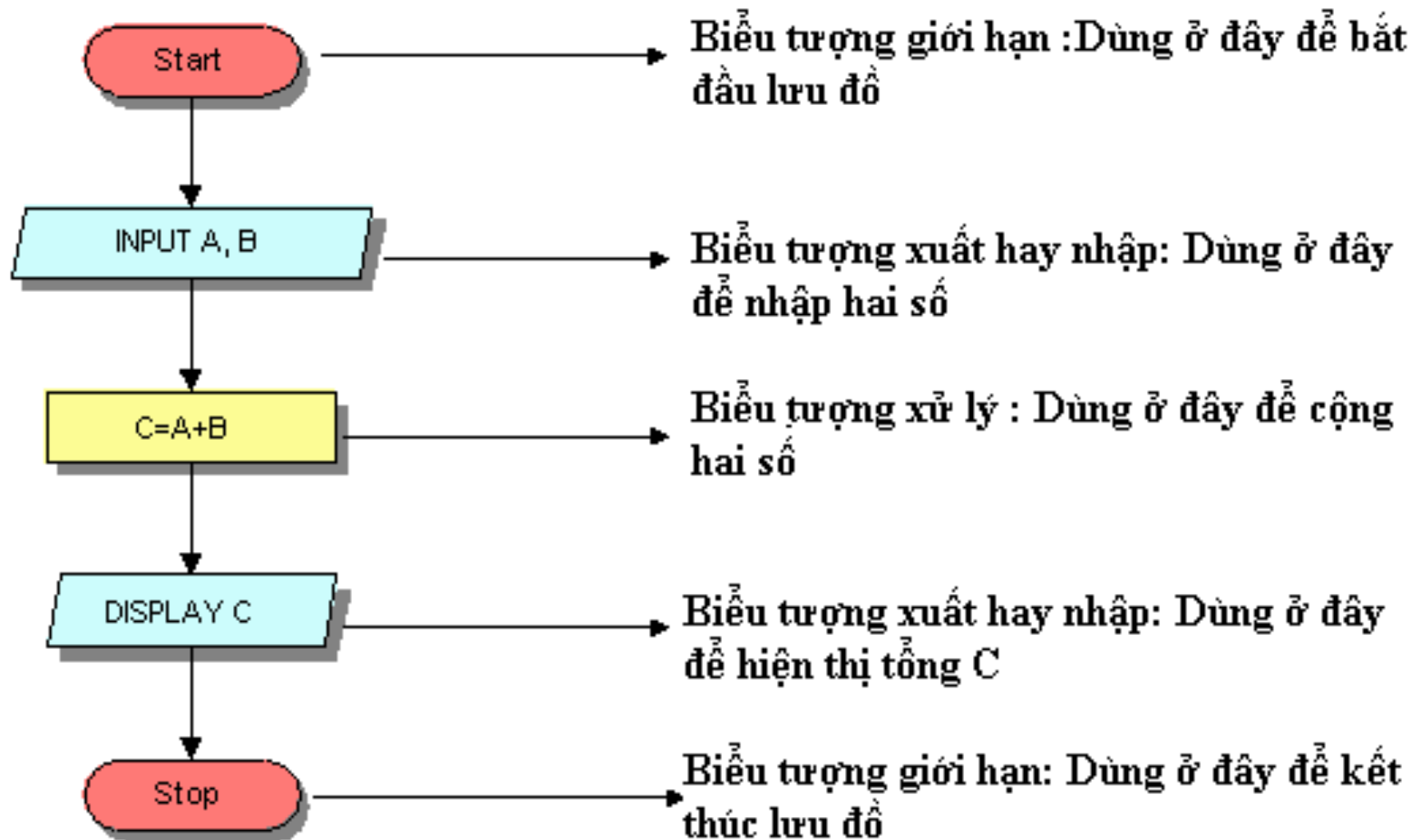
Lưu đồ là một hình ảnh minh họa cho giải thuật



Biểu tượng trong lưu đồ

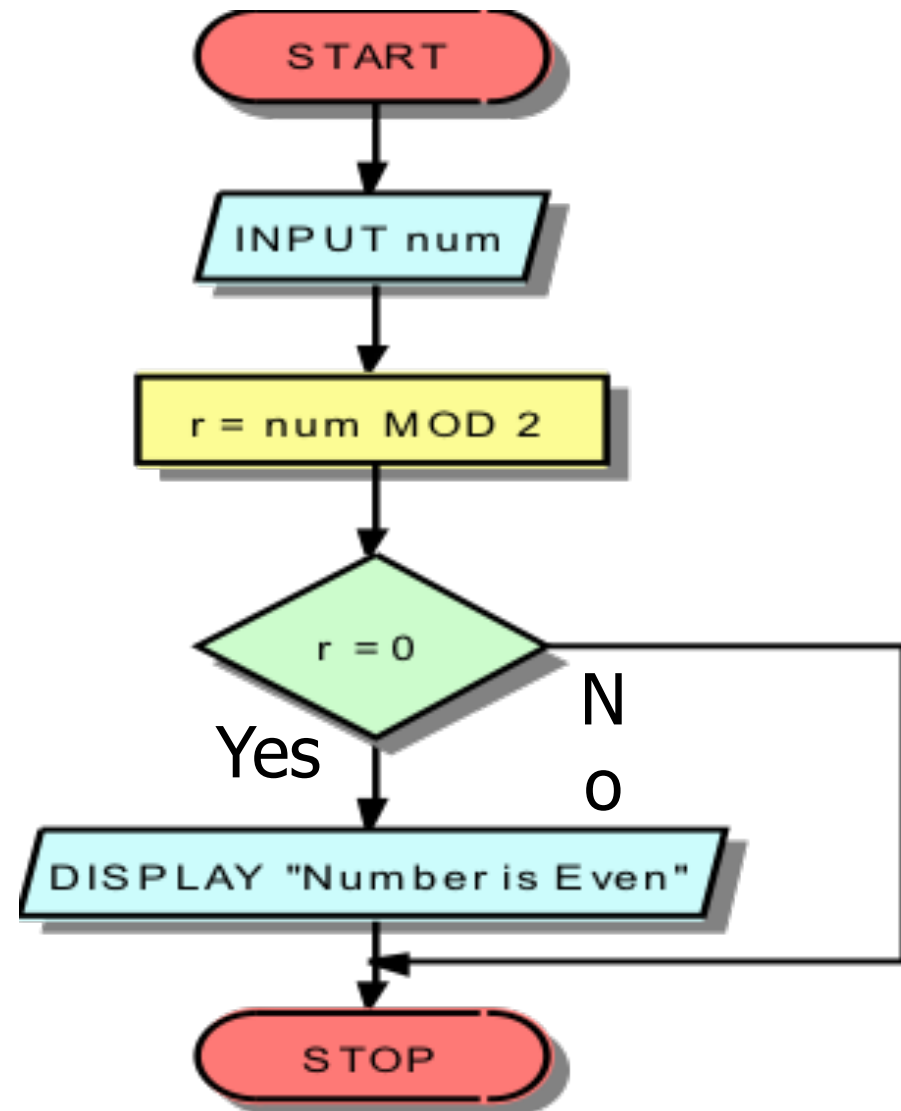
Biểu Tượng	Mô Tả
	Bắt đầu hay kết thúc chương trình
	Những bước tính toán
	Các lệnh xuất hay nhập
	Quyết định và rẽ nhánh
	Bộ nối hai phần trong chương trình (đầu nối)
	Dòng chảy

Lưu đồ cộng hai số



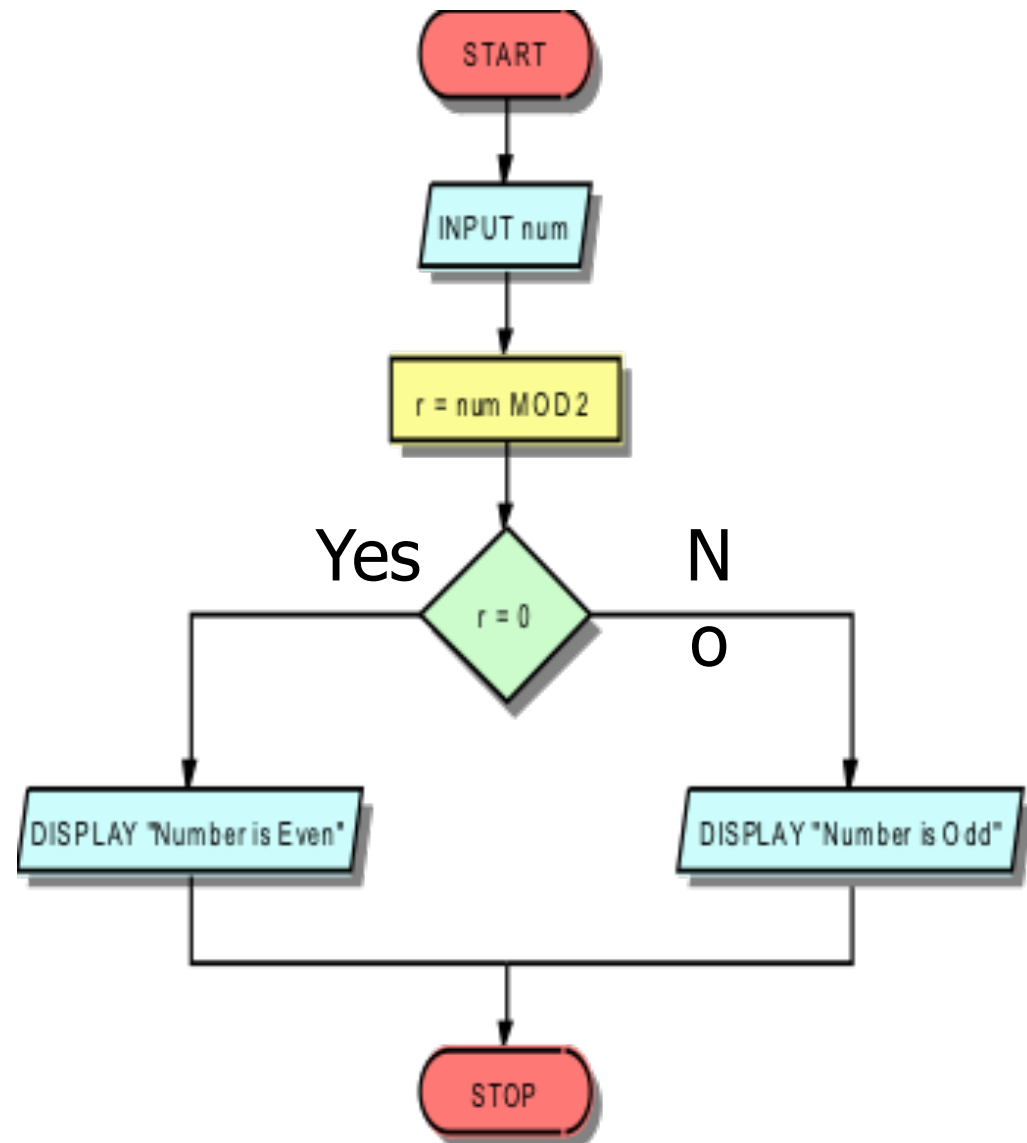
Cấu trúc IF

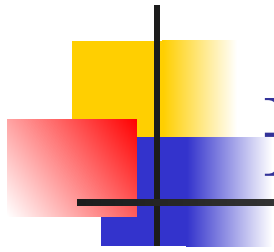
```
BEGIN  
INPUT num  
r = num MOD 2  
IF r=0  
    Display “Number is even”  
END IF  
END
```



Cấu trúc IF...ELSE

```
BEGIN  
  INPUT num  
  r=num MOD 2  
  IF r=0  
    DISPLAY "Even Number"  
  ELSE  
    DISPLAY "Odd Number"  
  END IF  
END
```





Đa điều kiện sử dụng AND/OR

BEGIN

INPUT yearsWithUs

INPUT bizDone

IF yearsWithUs \geq 10 **AND** bizDone \geq 5000000

 DISPLAY “Classified as an MVS”

ELSE

 DISPLAY “A little more effort required!”

END IF

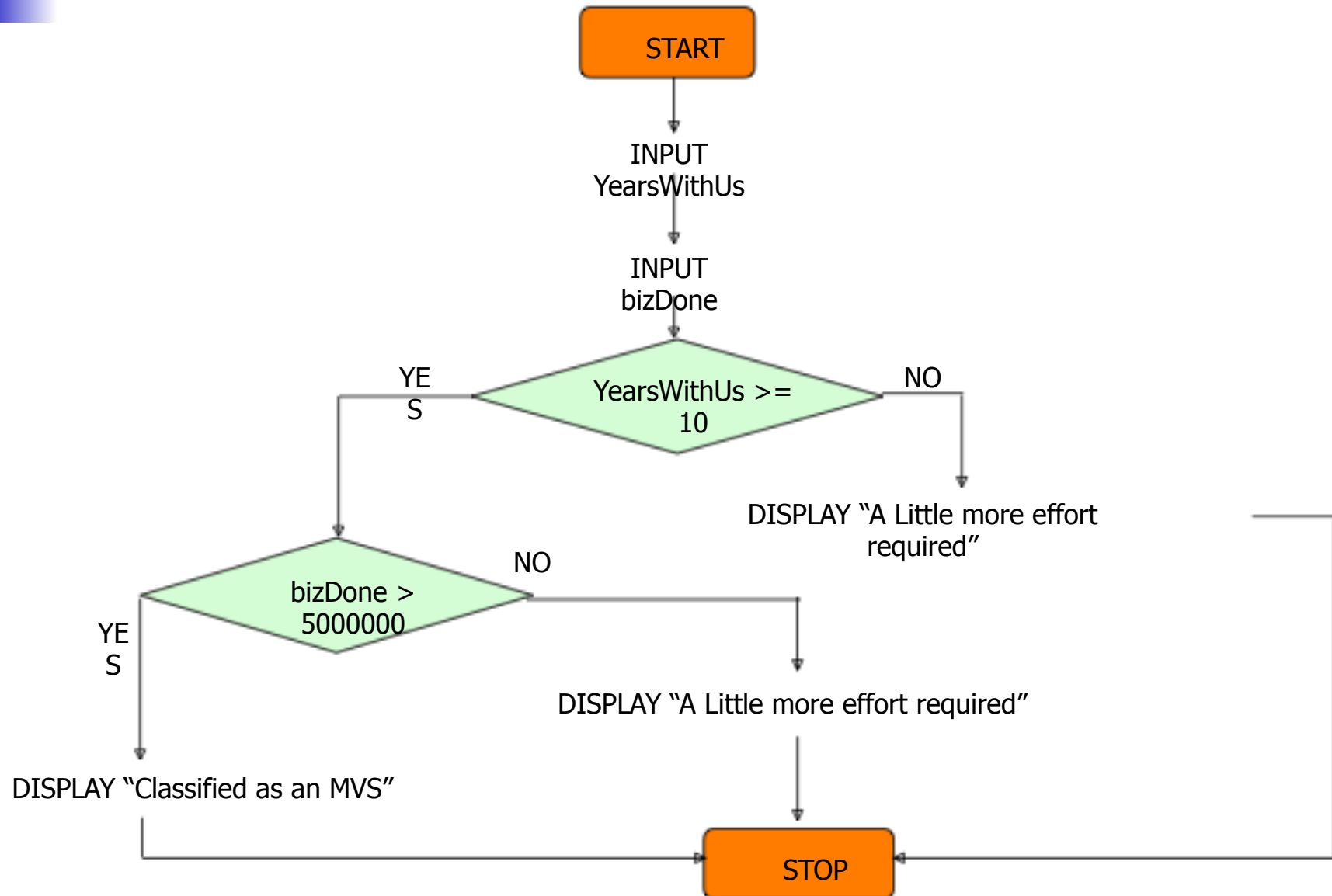
END



Cấu trúc IF lồng nhau

```
BEGIN
INPUT yearsWithUs
INPUT bizDone
IF yearsWithUs >= 10
    IF bizDone >= 5000000
        DISPLAY "Classified as an MVS"
    ELSE
        DISPLAY "A little more effort required!"
END IF
ELSE
    DISPLAY "A little more effort required!"
END IF
END
```


Cấu trúc IF lồng nhau (tt.)



Vòng lặp

```
BEGIN  
cnt=0  
WHILE (cnt < 1000)  
DO  
    DISPLAY "Scooby"  
    cnt=cnt+1  
END DO  
END
```

