



APTECH SAIGON

HỆ THỐNG ĐÀO TẠO LẬP TRÌNH VIỆN QUỐC TẾ

Lập trình C

- Các khái niệm căn bản
- Làm quen với ngôn ngữ C



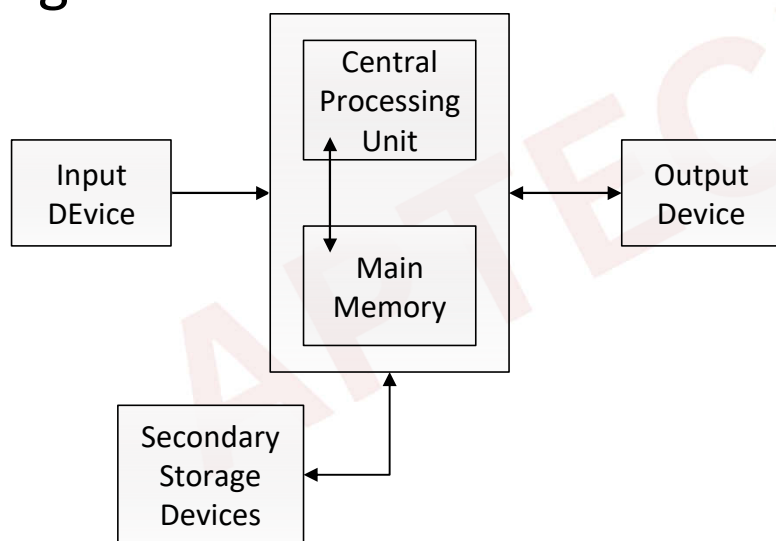
Các khái niệm cơ bản

1. Máy tính (computer):
 - Là một máy được thiết kế để thực hiện các lệnh chương trình (instructor, code).
2. Chương trình (program):
 - Là một chuỗi các lệnh nằm trong bộ nhớ máy tính để thực hiện công việc nào đấy.
3. Lập trình viên (programmer):
 - Là người viết ra các lệnh (chương trình) để máy tính thực hiện công việc

Hệ thống máy tính: Phần cứng, Phần Mềm

Phần cứng

Các thành phần chính của phần cứng



Phần Mềm

– Là các chương trình chạy trên máy tính.

– Các loại phần mềm (software)

- Hệ điều hành: Các chương trình dùng để quản lý phần cứng và các chương trình chạy trên máy tính. Ví dụ: Microsoft Windows, Unix, Linux,...
- Phần mềm ứng dụng: các chương trình cung cấp các dịch vụ sử dụng cho người dùng.

Chương trình và ngôn ngữ lập trình

1. Chương trình: tập hợp các lệnh máy tính để thực thi một tác vụ nào đó.
2. Ngôn ngữ lập trình: ngôn ngữ dùng để viết chương trình.
 - Các loại ngôn ngữ:
 - Ngôn ngữ cấp thấp: được sử dụng làm liên lạc giữa các bộ phận phần cứng. Thường được viết dưới dạng mã nhị phân.
 - Ngôn ngữ cấp cao: gần gũi với ngôn ngữ tự nhiên của con người.
 - C là ngôn ngữ cấp thấp hay cấp cao?

Chương trình nguồn

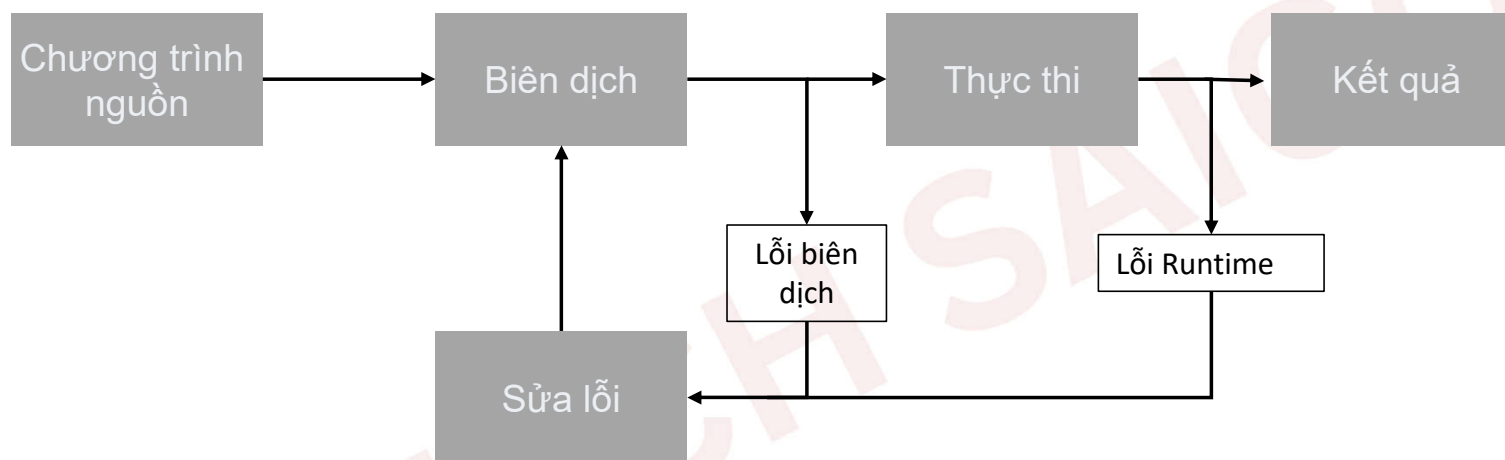
Chương trình (chương trình nguồn) gồm các phần tử của ngôn ngữ lập trình như:

- Từ khóa: còn gọi là các từ dành riêng của ngôn ngữ
- Các danh hiệu: kí hiệu do người lập trình định nghĩa.
- Các toán tử: sử dụng để thực hiện các tác vụ trên dữ liệu.
- Các kí tự phân cách hoặc kết thúc.
- Cú pháp của ngôn ngữ lập trình: gồm các quy tắc ngữ pháp của ngôn ngữ lập trình phải tuân theo khi viết chương trình. Điều khiển việc sử dụng các thành phần như: từ khóa, toán tử, danh hiệu và ký tự phân cách.
- Ví dụ

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

```
int main() {  
    printf("Hello World!");  
    return 0;  
}
```

Quá trình thực thi một chương trình nguồn



Các bước thực hiện của chương trình

- Nhận dữ liệu vào: Từ bàn phím, chuột, các thiết bị ngoại vi, Từ các tập tin trên đĩa, dữ liệu mạng,...
- Xử lý dữ liệu đầu vào: công việc chính của chương trình.
- Hiển thị kết quả ở đầu ra: Màn hình, thiết bị ngoại vi, các tập tin, mạng,...

Quy trình xây dựng chương trình

Giai đoạn 1: (Xác định mục tiêu). Cần xác định rõ đầu vào và đầu ra của chương trình

Giai đoạn 2: (Thiết kế và xây dựng chương trình) là công đoạn quan trọng nhất. Cần phân tích kỹ các yêu cầu (về mục tiêu, giao diện, về khả năng hiệu quả của chương trình...) để thiết kế cơ sở dữ liệu, thuật toán, giao diện...cho phù hợp

Giai đoạn 3: (Sửa lỗi và kiểm tra tính hiệu quả): Là giai đoạn kiểm tra và sửa lỗi phát sinh trong quá trình sử dụng (testing) và đánh giá hiệu quả của chương trình so với mục tiêu đề ra ban đầu

Lập trình thủ tục và lập trình hướng đối tượng

- Lập trình thủ tục: tập trung vào các quá trình, tác vụ. Chương trình được viết dưới dạng thủ tục (hàm) để xử lý các tác vụ, xác định các công việc phải làm để giải quyết vấn đề.
- Lập trình hướng đối tượng: tập trung vào các đối tượng của chương trình. Đối tượng bao gồm dữ liệu và các công cụ dùng để xử lý dữ liệu. Dùng thông điệp (message) để gửi các tác vụ yêu cầu tới đối tượng.



Làm quen với ngôn ngữ C

- Tại sao C
- Cấu trúc 1 chương trình C
- Các từ khóa
- Lệnh vào ra
- Kiểu dữ liệu
- Biến, hằng, biểu thức
- PSEUDOCODE



Cấu trúc 1 chương trình C

Chương trình	Các thành phần của chương trình
<pre>#include <stdio.h></pre>	Khai báo các hàm thư viện
<pre>int Sum(int a, int b); int s; int main() { int a, b; printf("Nhap a"); scanf("%d", &a); printf("Nhap b"); scanf("%d", &b); s = Sum(a, b); printf("Tong %d + %d=%d", a, b, s); }</pre>	Phần khai báo các biến và các hàm dùng chung
<pre>int Sum(int x, int y) { return x+y; }</pre>	Phần chương trình chính
	Phần định nghĩa các hàm dùng chung



Các từ khóa của C

- Có 32 từ khóa
- Tất cả từ khóa sử dụng chữ thường

Auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

- Không được sử dụng các keywords trên để đặt tên biến, hàm.

Pseudocode (Pseudo code mã giả)

- Sử dụng những quy ước có cấu trúc của một số ngôn ngữ lập trình, nhưng thường bỏ đi những chi tiết không cần thiết để giúp hiểu rõ giải thuật hơn.
- Mục đích của việc sử dụng mã giả là nó có thể con người đọc hiểu dễ dàng hơn những ngôn ngữ lập trình thông thường.

Code C	PseudoCode
<pre>if (subject=='C') { printf("Mon hoc dau tien"); } else { printf("Khong phai mon hoc dau tien") }</pre>	<pre>if subject=C then Hiển thị: Môn học đầu tiên else Hiển thị: Không phải môn học đầu tiên end if</pre>

Flowcharts (biểu đồ)

- Biểu diễn đồ họa của một giải thuật thay thế pseudocode. Là Công cụ biểu diễn thuật toán, mô tả nhập, xuất và luồng xử lý thông qua các ký hiệu hình học
- Sử dụng những hình khối đi kèm mũi tên chỉ dẫn. Cấu trúc bên trong sẽ bao gồm các bước, thứ tự giải quyết vấn đề của bài toán.
- Các ký hiệu quy ước

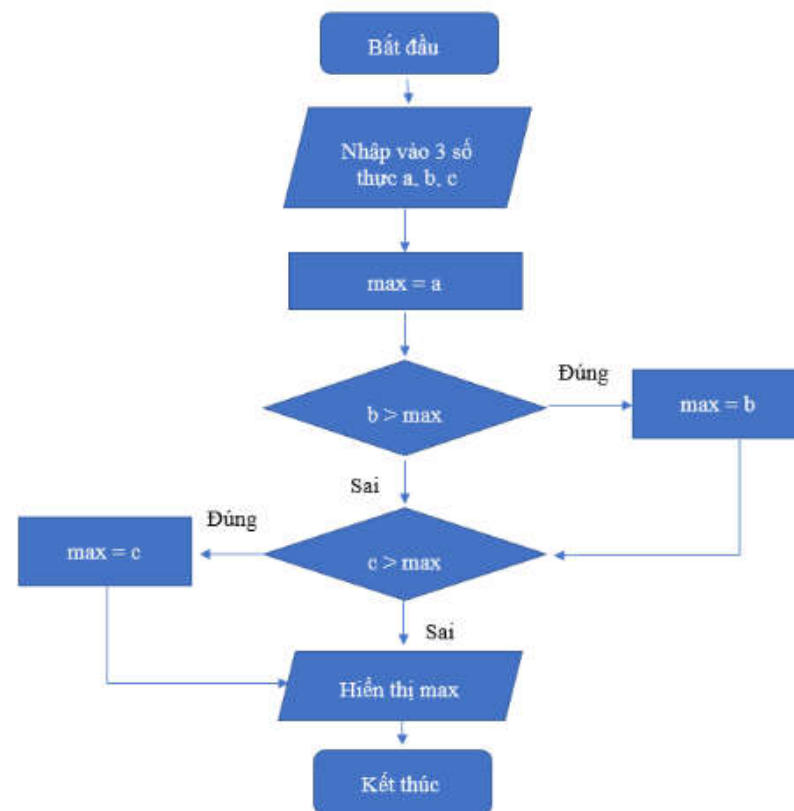
STT	Ký hiệu	Ý nghĩa
1		Bắt đầu / Kết thúc chương trình
2		Điều kiện rẽ nhánh (lựa chọn)
3		Luồng xử lý
4		Nhập
5		Xuất
6		Xử lý / tính toán / gán
7		Trả về giá trị (return)
8		Điểm nối liên kết

Flowchart – các ví dụ

- Nhập và tính tổng điểm và điểm trung bình

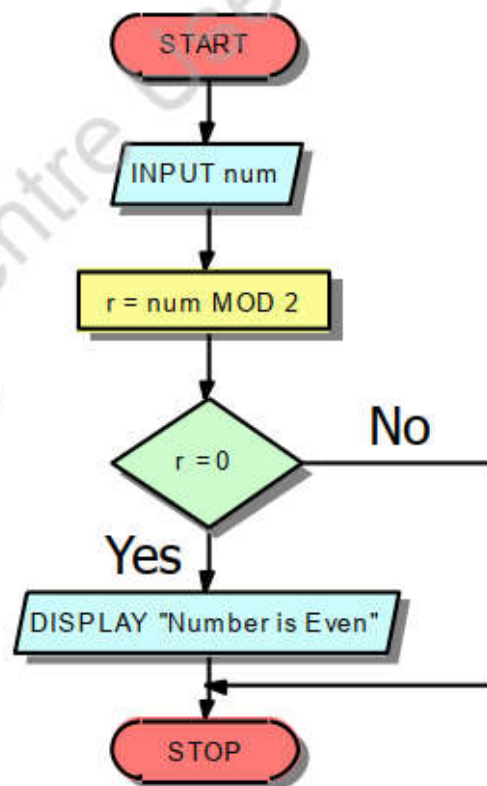


- Nhập 3 số và tìm số lớn nhất



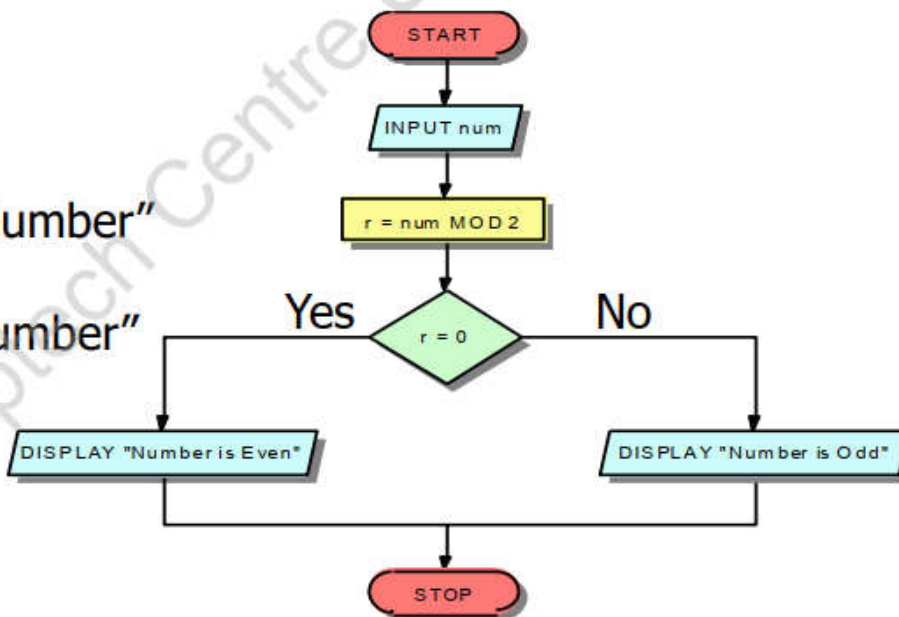
Flowchart và Pseudocode (ví dụ)

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

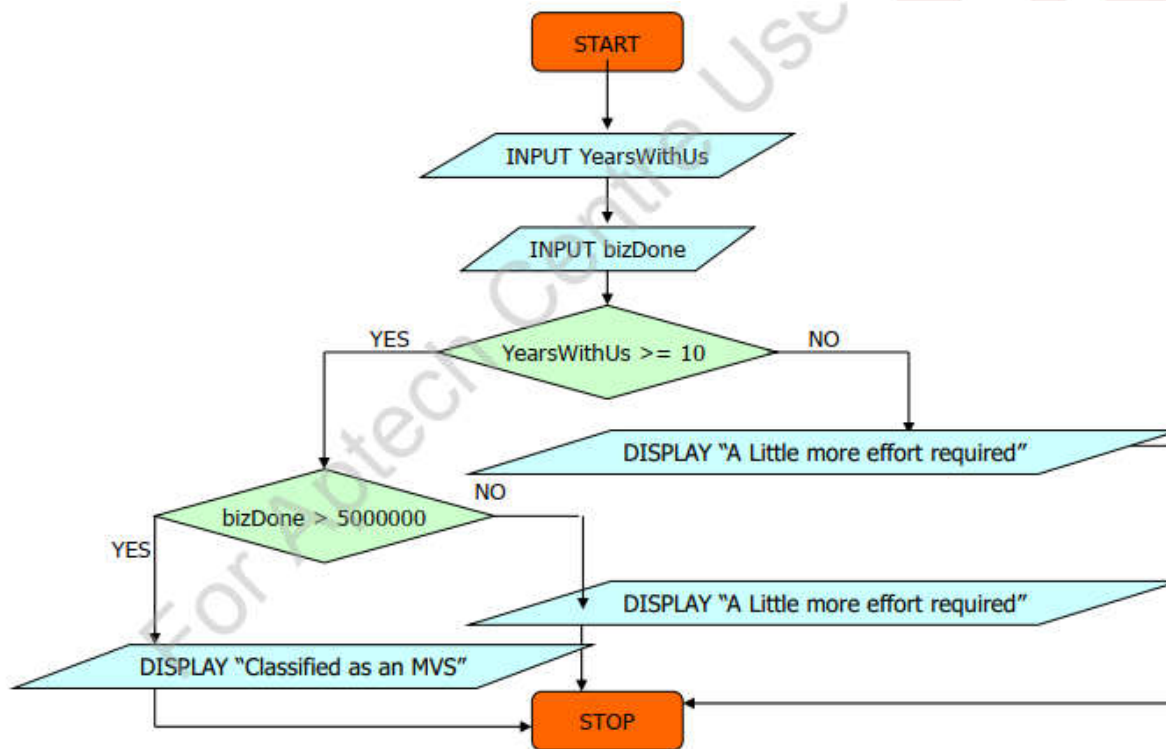


Flowchart và Pseudocode (ví dụ)

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



Flowchart và Pseudocode (ví dụ)



Flowchart và Pseudocode (ví dụ)

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

