

Các bước lập trình để giải quyết vấn đề

Chúng ta thường gặp phải những bài toán. Để giải quyết những bài toán đó, chúng ta cần hiểu chúng trước rồi sau đó mới hoạch định các bước cần làm.

Giả sử chúng ta muốn đi từ phòng học đến quán ăn tự phục vụ ở tầng hầm. Để thực hiện việc này chúng ta cần hiểu nó rồi tìm ra các bước giải quyết trước khi thực thi các bước đó:

1. Rời phòng
2. Đến cầu thang
3. Xuống tầng hầm
4. Đi tiếp đến quán ăn tự phục vụ

Thủ tục trên liệt kê tập hợp các bước thực hiện được xác định rõ ràng cho việc giải quyết vấn đề. Một tập hợp các bước như vậy gọi là giải thuật (Algorithm hay gọi vắn tắt là algo).

Một giải thuật (còn gọi là thuật toán) có thể được định nghĩa như là một thủ tục, công thức hay cách giải quyết vấn đề. Nó gồm một tập hợp các bước giúp đạt được lời giải.

Qua phần trên, chúng ta thấy rõ ràng để giải quyết được một bài toán, trước tiên ta phải hiểu bài toán đó, kể đến tập hợp tất cả những thông tin liên quan tới nó, bước kế sẽ là xử lý những mẫu thông tin đó, cuối cùng, chúng ta cho ra lời giải.

Giải thuật chúng ta có là một tập hợp các bước được liệt kê dưới dạng ngôn ngữ đơn giản. Rất có thể rằng các bước trên do hai người khác nhau viết vẫn tương tự nhau nhưng ngôn ngữ dùng diễn tả các bước có thể khác nhau. Do đó, cần thiết có những phương pháp chuẩn mực cho việc viết giải thuật để mọi người dễ dàng hiểu. Chính vì vậy , giải thuật được viết bằng cách dùng hai phương pháp chuẩn là mã giả (pseudo code) và lưu đồ (flowchart).

Cả hai phương pháp này đều dùng để xác định một tập hợp các bước cần được thi hành để có được lời giải. Liên hệ tới vấn đề đi đến quán ăn tự phục vụ trên, chúng ta đã vạch ra một kế hoạch (thuật toán) để đến đích. Tuy nhiên, để đến nơi, chúng ta phải cần thi hành những bước này thật sự. Tương tự, mã giả và lưu đồ chỉ đưa ra những bước cần làm. Lập trình viên phải viết mã cho việc thực thi những bước này qua việc dùng một ngôn ngữ nào đó.

Chi tiết về mã giả và lưu đồ được trình bày dưới đây.

1. Mã giả (pseudo code)

Nhớ rằng mã giả không phải là mã thật. Mã giả sử dụng một tập hợp những từ tương tự như mã thật nhưng nó không thể được biên dịch và thực thi như mã thật.

Chúng ta hãy xem xét mã giả qua ví dụ sau. Ví dụ này sẽ hiển thị câu Hello World!.

Ví dụ 1:

```
BEGIN  
DISPLAY 'Hello World!'  
END
```

Qua ví dụ trên, mỗi đoạn mã giả phải bắt đầu với từ BEGIN hoặc START, và kết thúc với từ END hay STOP. Để hiển thị giá trị nào đó, từ DISPLAY hoặc WRITE được dùng. Khi giá trị được hiển thị là một giá trị hằng (không đổi), trong trường hợp này là "Hello World", nó được đặt bên trong dấu nháy. Tương tự, để nhận một giá trị của người dùng, từ INPUT hay READ được dùng.

Để hiểu điều này rõ hơn, chúng ta xem xét ví dụ 2, ở ví dụ này ta sẽ nhập hai số và máy sẽ hiển thị tổng của hai số.

Ví dụ 2:

```
BEGIN  
INPUT A, B
```

```
DISPLAY A + B  
END
```

Trong đoạn mã giả này, người dùng nhập vào hai giá trị, hai giá trị này được lưu trong bộ nhớ và có thể được truy xuất như là A và B theo thứ tự. Những vị trí được đặt tên như vậy trong bộ nhớ gọi là biến. Chi tiết về biến sẽ được giải thích trong phần sau của chương này. Bước kế tiếp trong đoạn mã giả sẽ hiển thị tổng của hai giá trị trong biến A và B.

Tuy nhiên, cũng đoạn mã trên, ta có thể bổ sung để lưu tổng của hai biến trong một biến thứ ba rồi hiển thị giá trị biến này như trong ví dụ 3 sau đây.

Ví dụ 3:

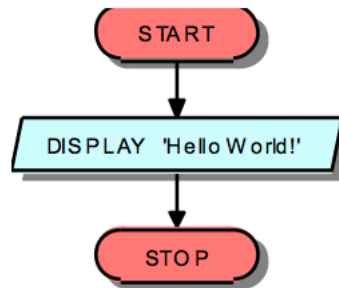
```
BEGIN  
INPUT A, B  
C = A + B  
DISPLAY C  
END
```

Một tập hợp những chỉ thị hay các bước trong mã giả thì được gọi chung là một cấu trúc. Có ba loại cấu trúc : tuần tự, chọn lựa và lặp lại. Trong đoạn mã giả ta viết ở trên, chúng ta dùng cấu trúc tuần tự. Chúng được gọi như vậy vì những chỉ thị được thi hành tuần tự, cái này sau cái khác và bắt đầu từ điểm đầu tiên. Hai loại cấu trúc còn lại sẽ được đề cập trong những chương sau.

Lưu đồ (Flowcharts)

Một lưu đồ là một hình ảnh minh họa cho giải thuật. Nó vẽ ra biểu đồ của luồng chỉ thị hay những hoạt động trong một tiến trình. Mỗi hoạt động như vậy được biểu diễn qua những ký hiệu.

Để hiểu điều này rõ hơn, chúng ta xem lưu đồ trong hình 1.1 dùng hiển thị thông điệp truyền thông Hello World!.



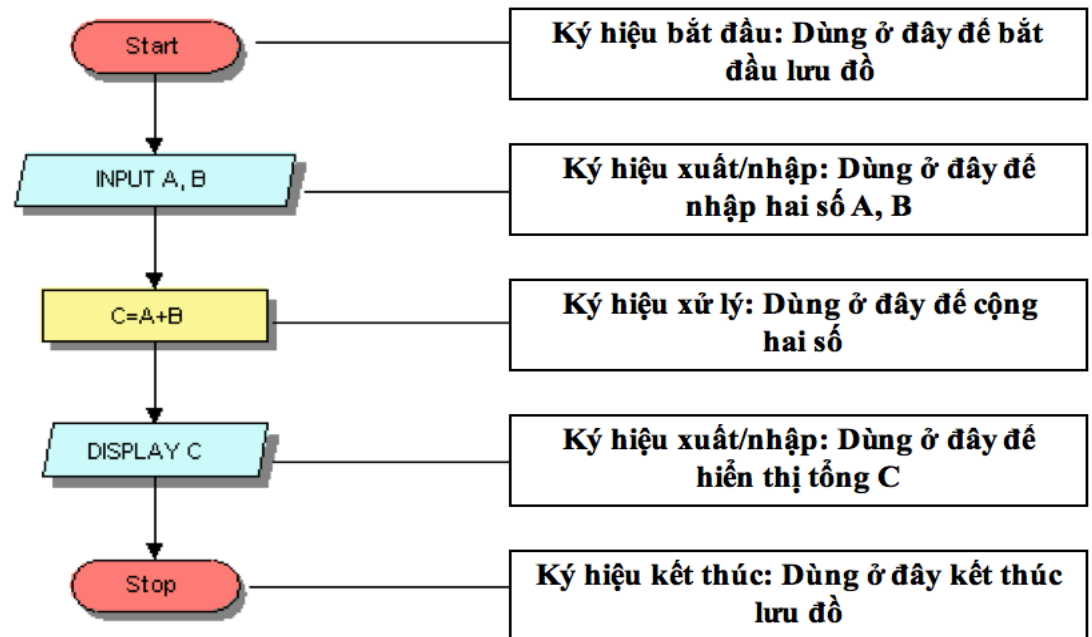
Hình 1.1. Hiển thị thông điệp

Lưu đồ giống với đoạn mã giả là cùng bắt đầu với từ BEGIN hoặc START, và kết thúc với từ END hay STOP. Tương tự, từ khóa DISPLAY được dùng để hiển thị giá trị nào đó đến người dùng. Tuy nhiên, ở đây, mọi từ khóa thì nằm trong những ký hiệu. Những ký hiệu khác nhau mang một ý nghĩa tương ứng được trình bày ở bảng trong Hình 1.2.

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

Hình 1.2. Ký hiệu trong lưu đồ

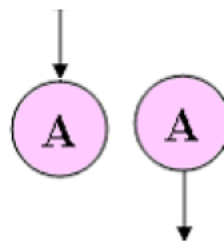
Ta hãy xét lưu đồ cho ví dụ 3(bài toán cộng hai số) như ở Hình 1.3 dưới đây:



Hình 1.3. lưu đồ cộng hai số

Tại bước mà giá trị của hai biến được cộng và gán cho biến thứ ba thì xem như là một xử lý và được trình bày bằng một hình chữ nhật.

Lưu đồ mà chúng ta xét ở đây là đơn giản. Thông thường, lưu đồ trải rộng trên nhiều trang giấy. Trong trường hợp như thế, biểu tượng bộ nối được dùng để chỉ điểm nối của hai phần trong một chương trình nằm ở hai trang kế tiếp nhau. Vòng tròn chỉ sự nối kết và phải chứa ký tự hoặc số như ở hình 1.4. Như thế, chúng ta có thể tạo liên kết giữa hai lưu đồ chưa hoàn chỉnh.



Hình 1.4. Liên kết hai lưu đồ

Bởi vì lưu đồ được sử dụng để viết chương trình, chúng cần được trình bày sao cho mọi lập trình viên hiểu chúng dễ dàng. Nếu có ba lập trình viên dùng ba ngôn ngữ lập trình khác nhau để viết mã, bài toán họ cần

giải quyết phải như nhau. Trong trường hợp này, mã giả đưa cho lập trình viên có thể giống nhau mặc dù ngôn ngữ lập trình họ dùng và tất nhiên là cú pháp có thể khác nhau. Nhưng kết quả cuối cùng là một. Do đó, cần thiết phải hiểu rõ bài toán và mã giả phải được viết cẩn thận. Chúng ta cũng kết luận rằng mã giả độc lập với ngôn ngữ lập trình.

Vài điểm cần thiết khác phải chú ý khi vẽ một lưu đồ :

- Lúc đầu chỉ tập trung vào khía cạnh logic của bài toán và vẽ các luồng xử lý chính của lưu đồ
- Một lưu đồ phải có duy nhất một điểm bắt đầu START và một điểm kết thúc STOP.
- Không cần thiết phải mô tả từng bước của chương trình trong lưu đồ mà chỉ cần các bước chính và có ý nghĩa cần thiết.