

# 1. Tổng quan về lập trình

---

[GitHub: k1enn](#)[Codeforces: dinhtrungkien](#)[LinkedIn: Trung Kiên](#)

## Nội dung

- Các khái niệm cơ bản, sẽ kết hợp với tiếng Anh để các bạn dễ tìm tài liệu.
- Các phương pháp mô tả thuật toán (Algorithm)

## Trình biên dịch

---

**Trình biên dịch:** Giúp dịch chương trình viết bằng ngôn ngữ lập trình bậc cao (high-level programming language) sang ngôn ngữ lập trình bậc thấp.

Có rất nhiều ngôn ngữ lập trình khác nhau. Tùy vào nhu cầu, ngành học sẽ học ngôn ngữ lập trình cho phù hợp.

HUFLIT hiện đang dạy C# cho các môn nền tảng như Cấu trúc Dữ liệu và Giải thuật, Kỹ thuật Lập trình và dạy theo .NET Framework cho môn Lập trình Web, các bạn sẽ được học sau.

## Chương trình máy tính

---

**Chương trình máy tính** (Computer program) là tập hợp các câu lệnh viết bằng ngôn ngữ lập trình (programming language) để chỉ dẫn máy tính giải quyết vấn đề theo yêu cầu.

## Lập trình viên

---

Lập trình viên (Programmer) là người sử dụng ngôn ngữ lập trình để viết ra chương trình máy tính.

## Bài toán tin học

---

Gồm 3 thành phần:

- Input
- Output
- Constraint

Các dạng chính:

- Decision
- Search

- Counting
- Optimization

Phần này các bạn học tiếp sẽ hiểu thêm, 3 thành phần chính sẽ được nhắc đi nhắc lại suốt 4 năm học.

## Thuật toán

---

**Thuật toán** (Algorithm) là một dãy các câu lệnh nhất định để máy tính từng bước giải quyết vấn đề.

Các đặc trưng của thuật toán:

- Tính Xác định: Cụ thể hóa các bước
- Tính Hữu hạn: Thuật toán có điểm dừng, không thể chạy vô hạn
- Tính Chính xác: Kết quả chính xác
- Tính Hiệu quả: Sử dụng càng ít tài nguyên càng tốt
- Tính Phổ quát: Áp dụng được nhiều trường hợp cho một dạng toán

## 3 cấu trúc cơ bản

---

Thông thường một chương trình sẽ kết hợp các cấu trúc này với nhau, nhưng với chương trình cơ bản thì thường sẽ chỉ có từ 1-2 cấu trúc được sử dụng.

### Tuần tự

Cơ bản nhất, code chạy từ trên xuống

Thuật toán tính và in ra tổng 2 số nguyên  $a$ ,  $b$

- Bước 1: Nhập  $a$ ,  $b$
- Bước 2: Tính  $sum = a + b$
- Bước 3: In  $sum$
- Bước 4: Kết thúc.

### Lựa chọn

Xét giá trị, trả về `true/false` để lựa chọn đoạn lệnh nào sẽ chạy.

Thuật toán kiểm tra  $n$  là số nguyên hay không

- Bước 1: Nhập  $n$
- Bước 2: Khởi tạo  $f = false$

- Bước 3:  
Nếu  $n > 0$ ,  $f = \text{true}$ . Sang bước 4  
Ngược lại,  $f = \text{false}$ . Sang bước 4
- Bước 4: In  $f$  tùy theo  $\text{true}$  hoặc  $\text{false}$
- Bước 5: Kết thúc

## Lặp

Thực hiện một hoặc nhiều dòng lệnh lặp theo số lần nhất định, lặp vô hạn hoặc vô định là sai. Khi học Kỹ thuật Lập trình sẽ có một cấu trúc tương tự là **Đệ quy**.

Thuật toán tính tổng:

$$S = 1 + 2 + \dots + n \quad (n > 0)$$

Bước 1: Nhập  $n$

Bước 2: Khởi tạo  $S = 0$ ,  $i = 1$

Bước 3: Nếu  $i \leq n$ ,  $S = S + i$ . Chuyển sang bước 4; Ngược lại, chuyển sang bước 5;

Bước 4:  $i = i + 1$ . Quay lại bước 3

Bước 5: In  $S$ .

Bước 6: Kết thúc

Tham khảo dãy Fibonacci nếu bạn muốn hiểu hơn về sử dụng Đệ quy.

## Các phương pháp mô tả thuật toán

3 phương pháp mô tả thuật toán:

- Ngôn ngữ tự nhiên: sử dụng ngôn ngữ thông thường để mô tả các bước của thuật toán (như các ví dụ ở phần trước)
- Mã giả (pseudocode): vay mượn từ khóa và cú pháp của một ngôn ngữ lập trình nào đó để mô tả thuật toán
- Sơ đồ khối (flowchart): sử dụng các hình khối để biểu các thao tác và trình tự thực hiện theo một quy ước chung

Thông thường khi học giải thuật trên **Leetcode** các bạn sẽ gặp các phương pháp mô tả này.

Với quy mô nhỏ hơn khi học năm nhất, **cố gắng suy nghĩ thuật toán** để mô tả bằng những cách này. Sau đó hãy bắt đầu code.