

## BÀI TẬP BUỔI 1 – KỸ THUẬT LẬP TRÌNH

Tên / MSSV : Phạm Đức Trí – 074206002542

### Câu 1: Trình bày theo ý nghĩ của riêng “ban” tại sao ban học lập trình ?

Là một sinh viên ngành công nghệ kỹ thuật điều khiển và tự động hóa, mình nhận thấy việc học lập trình là một phần rất quan trọng trong quá trình học tập cũng như định hướng nghề nghiệp của bản thân. Bản thân mình muốn học lập trình vì mình tin rằng nó không chỉ giúp mình hiểu sâu hơn về các hệ thống tự động, mà còn mở ra nhiều cơ hội phát triển trong tương lai.

Cụ thể, việc học lập trình mang lại cho mình 3 lợi ích và giá trị lớn sau:

1. **Tư duy logic và giải quyết vấn đề:** Lập trình giúp mình rèn luyện cách suy nghĩ có hệ thống, từ đó mình có thể phân tích và đưa ra giải pháp cho những vấn đề phức tạp trong học tập cũng như trong cuộc sống.
2. **Ứng dụng trực tiếp trong chuyên ngành:** Trong lĩnh vực tự động hóa, lập trình là công cụ không thể thiếu để thiết kế, điều khiển và tối ưu các hệ thống. Việc thành thạo lập trình sẽ giúp mình dễ dàng tiếp cận các dự án thực tế, từ điều khiển robot đến phát triển hệ thống nhúng.
3. **Cơ hội nghề nghiệp và phát triển bản thân:** Lập trình mở ra nhiều cơ hội việc làm trong cả lĩnh vực công nghệ thông tin lẫn tự động hóa. Ngoài ra, kỹ năng này còn giúp mình tự tin hơn trong việc nghiên cứu, sáng tạo và khởi nghiệp trong tương lai.

### Câu 2: Trình bày cách máy tính thực thi lệnh

Máy tính không hiểu trực tiếp ngôn ngữ lập trình bậc cao mà chỉ hiểu **ngôn ngữ máy** (mã nhị phân 0 và 1). Vì vậy, khi lập trình viên viết chương trình bằng các ngôn ngữ như C, Python, Java..., chương trình sẽ được **biên dịch (compile)** hoặc **thông dịch (interpret)** thành mã máy. Sau đó, CPU sẽ đọc và thực hiện từng lệnh theo một chu trình cơ bản gồm:  **nạp lệnh → giải mã lệnh → thực thi → lưu kết quả**.

Cụ thể chu trình này diễn ra như sau:

1. **Nạp lệnh (Fetch):** CPU lấy lệnh từ bộ nhớ chính dựa vào địa chỉ trong thanh ghi bộ đếm chương trình (PC), rồi đưa lệnh đó vào thanh ghi lệnh (IR). Sau khi nạp, PC tăng lên để trỏ đến lệnh tiếp theo.
2. **Giải mã lệnh (Decode):** Bộ điều khiển (CU) phân tích lệnh vừa nạp để xác định loại thao tác cần thực hiện (số học, logic, truy xuất bộ nhớ, nhảy điều kiện...).

3. **Thực thi lệnh (Execute):** CPU tiến hành thực hiện thao tác theo lệnh, ví dụ tính toán số học/logic trong ALU, hoặc đọc/ghi dữ liệu từ bộ nhớ.
4. **Lưu kết quả (Store):** Kết quả được lưu lại vào thanh ghi hoặc ghi trở lại bộ nhớ chính để các lệnh sau có thể sử dụng.

### 3. Trình biên dịch và trình thông dịch là gì? Lập bảng phân loại các ngôn ngữ lập trình theo trình biên dịch và trình thông dịch.

- **Trình biên dịch (Compiler):** Chuyển toàn bộ mã nguồn thành mã máy trước khi chạy. Sau khi biên dịch xong, chương trình có thể chạy độc lập mà không cần mã nguồn. → Ví dụ: C, C++, Java (biên dịch sang bytecode).
- **Trình thông dịch (Interpreter):** Dịch và thực thi từng dòng lệnh ngay khi chạy chương trình, không tạo ra file thực thi riêng. → Ví dụ: Python, JavaScript, PHP.

Loại	Ngôn ngữ ví dụ
Biên dịch	C, C++, Go, Pascal
Thông dịch	Python, JavaScript, PHP
Kết hợp (biên dịch + thông dịch)	Java (biên dịch sang bytecode rồi JVM thông dịch), C# (.NET CLR)

```

File Edit Selection View Go Run Terminal Help
Lap_trinh_C_Py
Gradient_Descent.py Linear_Regression_library.py Buoi_1.py
KT_lap_trinh > Buoi_1.py
1 print('Hello World! Tôi là Phạm Đức Trí')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Lap_trinh_C_Py> & C:/Users/ACER/AppData/Local/Programs/Python/Python311/python.exe c:/Lap_trinh_C_Py/KT_lap_trinh/Buoi_1.py
Hello World! Tôi là Phạm Đức Trí
PS C:\Lap_trinh_C_Py> & C:/Users/ACER/AppData/Local/Programs/Python/Python311/python.exe c:/Lap_trinh_C_Py/KT_lap_trinh/Buoi_1.py
Hello World! Tôi là Phạm Đức Trí
PS C:\Lap_trinh_C_Py>

```