

## BÀI TẬP 1

**Câu 1:** Trình bày theo ý nghĩ riêng của ‘bạn’, tại sao bạn học lập trình ?

⇒ Lý do em học lập trình : Để hiểu chuyên sâu hơn về ngôn ngữ máy tính, hiểu rõ hơn về AI, để học căn bản cho chuyên ngành của mình và muốn sáng tạo thêm nhiều cái mới và hay bằng lập trình như game các mô hình tự động hóa

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

⇒ Gồm 3 bước chính :

1. Tìm nạp : bộ đếm chương trình lưu giữ địa chỉ của lệnh tiếp theo cần được thực thi. CPU sử dụng địa chỉ này để tìm nạp lệnh. Lệnh được chuyển vào một thanh ghi bên trong CPU là Thanh ghi Lệnh .
2. Giải mã : Bộ điều khiển của CPU sẽ phân tích mã nhị phân của lệnh. Lệnh máy tính bao gồm hai phần chính: Mã thao tác : Cho biết loại hành động cần thực hiện . Toán hạng: Chỉ định dữ liệu cần thao tác hoặc địa chỉ bộ nhớ/thanh ghi chứa dữ liệu đó. Bộ điều khiển sẽ chuyển mã lệnh thành một chuỗi các tín hiệu điều khiển để kích hoạt các thành phần phần cứng phù hợp trong CPU chuẩn bị cho bước Thực thi.
3. Thực thi : Thao tác dữ liệu: thực hiện các phép tính toán học hoặc phép tính logic  
Chuyển dữ liệu: Đọc dữ liệu từ RAM, thanh ghi hoặc ghi kết quả ngược lại vào RAM, thanh ghi. Điều khiển: Thay đổi giá trị của PC để "nhảy" đến một lệnh không theo thứ tự tuần tự

Các bước diễn ra theo chu trình cho đến khi máy tắt hoặc chương trình kết thúc.

**Câu 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

⇒ 1. Trình biên dịch

Trình biên dịch là một chương trình dịch toàn bộ mã nguồn của chương trình thành một tệp mã máy độc lập trước khi chương trình được chạy.

Quá trình: dịch mã -> tạo tệp thực thi -> thực thi.

Ưu điểm: Tốc độ chạy lệnh nhanh vì mã đã được dịch sẵn sang ngôn ngữ máy

Nhược điểm: Mất thời gian biên dịch ban đầu, tệp thực thi phụ thuộc vào kiến trúc phần cứng.

⇒ 2. Trình thông dịch Trình thông dịch là một chương trình dịch và thực thi mã nguồn từng dòng một ngay trong lúc chương trình đang chạy.

Quá trình: Đọc từng dòng -> dịch -> thực thi -> lặp lại.

Ưu điểm: Dễ dàng gỡ lỗi khi thực hiện chương trình, tính đa nền tảng cao

Nhược điểm: Tốc độ thực thi chậm hơn vì quá trình dịch diễn ra lặp lại mỗi khi chương trình chạy.

### ⇒ 3. Phân loại Ngôn ngữ Lập trình

Ngôn ngữ lập trình thường được phân loại dựa trên cách chúng được chuyển đổi và thực thi.

Tiêu chí	Ngôn ngữ Biên dịch	Ngôn ngữ Thông dịch	Ngôn ngữ Kết hợp
Cơ chế hoạt động	Dịch toàn bộ mã nguồn thành mã máy trước khi chạy.	Dịch và thực thi từng dòng.	Dịch mã nguồn thành Bytecode mã trung gian, sau đó mã được thông dịch hoặc dịch Just-In-Time sang mã máy.
Tốc độ thực thi	Rất nhanh (đã được tối ưu hóa).	Chậm hơn do phải dịch lại mỗi lần chạy.	Nhanh (gần bằng Compiled) nhờ tối ưu hóa JIT.
Tính đa nền tảng	Thấp.	Cao.	Cao.
Thời điểm báo lỗi	Báo tất cả lỗi cùng lúc sau khi biên dịch.	Báo lỗi ngay tại dòng lệnh bị lỗi, dừng chương trình.	Báo lỗi ở cả giai đoạn biên dịch Bytecode và giai đoạn thực thi.
Ví dụ	C, C++, Go, Swift, Rust.	Python, Ruby, PHP, JavaScript (ban đầu).	Java , C# , JavaScript .

**Câu 4 :** Viết chương trình in ra màn hình dòng chữ: “[Hello world!](#) Tôi là xxx”. Trong đó xxx là tên sinh viên

