

Hãy thực hiện các bước sau để kiểm chứng đầu ra từ AI:

1. Đặt câu hỏi cho AI (ChatGPT, Claude, v.v...):
"Viết một đoạn code kiểm tra số nguyên tố bằng ngôn ngữ bạn chọn (C/C++, Python, Java...)"
2. Lưu lại đoạn code AI tạo ra.
3. Chạy thử đoạn code bằng công cụ lập trình bạn đang học (IDE, compiler,...).
 - Nếu chạy đúng: mô tả kết quả và bạn đã kiểm chứng thế nào.
 - Nếu chạy sai: xác định lỗi, chỉnh sửa code, và ghi chú lại vì sao AI sai.
4. Viết ngắn gọn phần kết luận:
 - Bạn rút ra bài học gì?
 - Cần làm gì để không bị lệ thuộc hoặc hiểu sai khi dùng AI?

Câu hỏi cho AI :

"Viết một đoạn code kiểm tra số nguyên tố bằng ngôn ngữ bạn chọn (C/C++, Python, Java...)"

```
n = int(input("Nhập n: "))
isPrime = True

if n < 2:
    isPrime = False
else:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            isPrime = False
            break

if isPrime:
    print(n, "là số nguyên tố")
else:
    print(n, "không là số nguyên tố")
```

Đoạn code trên là ĐÚNG.

Nó sử dụng thuật toán kiểm tra số nguyên tố phổ biến và tối ưu:

- Xử lý trường hợp đặc biệt: Kiểm tra $n < 2$ trước.
- Tối ưu hóa: Chỉ kiểm tra các ước số từ 2 đến căn bậc hai của n . Nếu n có một ước số lớn hơn căn bậc hai của n , nó chắc chắn phải có một ước số nhỏ hơn căn bậc hai của n . Việc kiểm tra đến căn n giúp giảm đáng kể số lần lặp.
- Hiệu quả (Break): Lệnh `break` đảm bảo rằng ngay khi tìm thấy ước số đầu tiên, chương trình sẽ dừng kiểm tra, giúp tối ưu thời gian chạy.

Bài học rút ra:

- AI (ChatGPT) đã cung cấp một giải pháp không chỉ chính xác về mặt logic mà còn tối ưu về mặt hiệu suất (sử dụng căn bậc hai), cho thấy khả năng của AI trong việc đề xuất các thuật toán tốt.

Cần làm gì để không bị lệ thuộc hoặc hiểu sai khi dùng AI:

Kiểm tra thử : Không chỉ kiểm thử trường hợp thành công, mà còn phải kiểm thử các trường hợp mà thuật toán có thể thất bại (số âm, số 0, số 1).

Tự gỡ lại code (Refactoring): Sau khi hiểu, em nên thử tự viết lại hàm theo một cách khác hoặc chỉnh sửa tên biến để củng cố kiến thức và tránh việc copy-paste thụ động.