

## 1. Câu hỏi đặt cho AI

Viết một đoạn code kiểm tra số nguyên tố bằng ngôn ngữ C.

## 2. Đoạn code AI tạo ra

```
#include <stdio.h>

int main() {
    int n, i;
    int flag = 0;

    printf("Nhập một số nguyên: ");
    scanf("%d", &n);

    for(i = 2; i <= n; i++) { // Lỗi logic
        if(n % i == 0) {
            flag = 1;
            break;
        }
    }

    if(flag == 0)
        printf("%d là số nguyên tố\n", n);
    else
        printf("%d không phải là số nguyên tố\n", n);

    return 0;
}
```

## 3. Chạy thử đoạn code

- Sử dụng IDE C (Code::Blocks / Dev C++).

- Nhập số 5 → chương trình in ra: “5 không phải là số nguyên tố” → **sai**.

#### 4. Phân tích lỗi

- Vòng lặp kiểm tra từ  $i = 2$  đến  $i \leq n$  → luôn chia hết cho chính nó.
- Lỗi: Ai không xử lý đúng điều kiện kiểm tra số nguyên tố.

#### 5. Chính sửa code

```
#include <stdio.h>

int main() {
    int n, i;
    int flag = 0;

    printf("Nhập một số nguyên: ");
    scanf("%d", &n);

    if(n < 2) {
        printf("%d không phải là số nguyên tố\n", n);
        return 0;
    }

    for(i = 2; i <= n / 2; i++) { // Chỉ kiểm tra đến n/2
        if(n % i == 0) {
            flag = 1;
            break;
        }
    }

    if(flag == 0)
        printf("%d là số nguyên tố\n", n);
    else
        printf("%d không phải là số nguyên tố\n", n);
}
```

```
    return 0;  
}
```

- Kết quả sau chỉnh sửa:
  - Nhập 5 → “5 là số nguyên tố” ✓
  - Nhập 4 → “4 không phải là số nguyên tố” ✓

## 6. Kết luận & bài học

- **Bài học rút ra:**
  - AI là công cụ hỗ trợ, nhưng **không phải lúc nào cũng đúng**.
  - Cần **hiểu logic, kiểm tra thực tế** trước khi áp dụng code AI.
- **Cách tránh lệ thuộc hoặc hiểu sai:**
  - Luôn đọc và phân tích code AI trước khi chạy.
  - Chạy thử với nhiều dữ liệu kiểm thử.
  - So sánh với lý thuyết, sách giáo trình, hoặc tự viết lại code.