## Prompt 1 (Ngắn gọn)

Viết hàm C kiểm tra số nguyên tố, có chú thích từng bước, đầu vào là số nguyên dương n.

### Kết quả code 1

#include <stdio.h>  
  
// Hàm kiểm tra số nguyên tố  
int isPrime(int n) {  
 if (n < 2) return 0; // Số nhỏ hơn 2 không phải số nguyên tố  
 for (int i = 2; i \* i <= n; i++) {  
 if (n % i == 0) return 0; // Có ước số khác 1 và chính nó  
 }  
 return 1; // Là số nguyên tố  
}

## Prompt 2 (Dài và chi tiết)

Hãy viết một chương trình C có hàm kiểm tra xem một số nguyên dương n có phải là số nguyên tố hay không.   
Yêu cầu:   
- Viết chú thích chi tiết từng bước trong code.  
- Giải thích nguyên lý kiểm tra.  
- In ra kết quả cho người dùng biết.

### Kết quả code 2

#include <stdio.h>  
  
// Hàm kiểm tra số nguyên tố  
// Nguyên lý: một số nguyên dương n là số nguyên tố nếu nó > 1   
// và không chia hết cho bất kỳ số nào từ 2 đến căn bậc hai của n.  
int isPrime(int n) {  
 if (n <= 1) return 0; // Loại bỏ số <= 1  
 for (int i = 2; i \* i <= n; i++) {  
 if (n % i == 0) return 0; // Nếu chia hết => không phải số nguyên tố  
 }  
 return 1; // Là số nguyên tố  
}  
  
int main() {  
 int n;  
 printf("Nhap mot so nguyen duong: ");  
 scanf("%d", &n);  
  
 if (isPrime(n))  
 printf("%d la so nguyen to.\n", n);  
 else  
 printf("%d khong phai la so nguyen to.\n", n);  
  
 return 0;  
}

## So sánh & Phân tích

\*\*So sánh kết quả AI sinh code:\*\*  
  
1. \*\*Mức độ chi tiết:\*\*  
 - Prompt ngắn → AI chỉ tạo hàm `isPrime()` đơn giản, tập trung vào chức năng chính.  
 - Prompt dài → AI sinh thêm cả `main()` để nhập/xuất dữ liệu, có chú thích giải thích nguyên lý hoạt động.  
  
2. \*\*Chú thích và giải thích:\*\*  
 - Phiên bản 1: Chú thích ngắn, mô tả từng dòng code.  
 - Phiên bản 2: Chú thích kỹ hơn, bao gồm mô tả thuật toán và lý do kiểm tra tới căn bậc hai.  
  
3. \*\*Phạm vi đầu ra:\*\*  
 - Phiên bản 1: Phù hợp khi muốn dùng lại trong chương trình khác.  
 - Phiên bản 2: Phù hợp cho người học mới vì có phần tương tác với người dùng.  
  
4. \*\*Tổng kết:\*\*   
 - Prompt \*\*dài và chi tiết hơn\*\* → giúp AI hiểu rõ yêu cầu, sinh code đầy đủ, có phần giải thích và giao diện người dùng.   
 - Prompt \*\*ngắn\*\* → tạo code ngắn gọn, tập trung vào hàm chính, phù hợp cho mục tiêu kỹ thuật.