## Prompt 1 (Cơ bản)

Viết hàm C tính tổng các số lẻ nhỏ hơn n.

### Kết quả code 1

#include <stdio.h>  
  
int sumOdd(int n) {  
 int sum = 0;  
 for (int i = 1; i < n; i++) {  
 if (i % 2 != 0) {  
 sum += i;  
 }  
 }  
 return sum;  
}

## Prompt 2 (Chi tiết, có chú thích)

Viết hàm C tính tổng các số lẻ nhỏ hơn n.   
Yêu cầu:   
- Code dễ đọc, đặt tên biến rõ ràng.  
- Có chú thích cho từng dòng lệnh để người học mới hiểu.

### Kết quả code 2

#include <stdio.h>  
  
// Hàm tính tổng các số lẻ nhỏ hơn n  
int tinhTongSoLe(int gioiHan) {  
 int tong = 0; // Biến lưu tổng các số lẻ  
  
 // Duyệt qua tất cả các số từ 1 đến (gioiHan - 1)  
 for (int soHienTai = 1; soHienTai < gioiHan; soHienTai++) {  
 // Nếu là số lẻ thì cộng vào tổng  
 if (soHienTai % 2 != 0) {  
 tong += soHienTai;  
 }  
 }  
  
 return tong; // Trả về kết quả cuối cùng  
}  
  
int main() {  
 int n;  
 printf("Nhap mot so nguyen duong: ");  
 scanf("%d", &n);  
  
 printf("Tong cac so le nho hon %d la: %d\n", n, tinhTongSoLe(n));  
 return 0;  
}

## So sánh & Phân tích

\*\*So sánh kết quả AI sinh code:\*\*  
  
1. \*\*Độ rõ ràng của code:\*\*  
 - Phiên bản 1: Biến đặt tên ngắn (`sum`, `i`), dễ hiểu với người đã biết C nhưng khó cho người mới học.  
 - Phiên bản 2: Đặt tên biến cụ thể (`gioiHan`, `tong`, `soHienTai`), giúp hiểu rõ ý nghĩa từng bước.  
  
2. \*\*Chú thích:\*\*  
 - Phiên bản 1: Không có chú thích.  
 - Phiên bản 2: Có chú thích từng dòng giải thích mục đích từng phần code.  
  
3. \*\*Phạm vi:\*\*  
 - Phiên bản 1: Chỉ có hàm, không có phần nhập/xuất.  
 - Phiên bản 2: Có thêm `main()`, cho phép người dùng nhập dữ liệu và xem kết quả.  
  
4. \*\*Tổng kết:\*\*   
 - Khi thêm yêu cầu “code dễ đọc, có chú thích dòng lệnh”, AI tạo code thân thiện, có tên biến rõ ràng và hướng đến người học mới.   
 - Phiên bản đầu chỉ phù hợp cho người đã quen ngôn ngữ C, còn phiên bản thứ hai dễ hiểu và đầy đủ hơn.