## Code gốc (viết rối, khó đọc)

#include <stdio.h>  
int main(){  
 int a,b=0,c,d,e;  
 printf("Nhap n: ");  
 scanf("%d",&a);  
 for(c=2;c<=a;c++){  
 int f=0;  
 for(d=2;d<=c/2;d++){  
 if(c%d==0){  
 f=1;  
 break;  
 }  
 }  
 if(f==0)  
 b=b+c;  
 }  
 printf("Tong cac so nguyen to <= n la: %d",b);  
 return 0;  
}

## Code sau khi tái cấu trúc (refactor)

#include <stdio.h>  
#include <stdbool.h>  
  
// Hàm kiểm tra số nguyên tố  
bool laSoNguyenTo(int n) {  
 if (n < 2) return false;  
 for (int i = 2; i \* i <= n; i++) {  
 if (n % i == 0) return false;  
 }  
 return true;  
}  
  
// Hàm chính: tính tổng các số nguyên tố ≤ n  
int main() {  
 int n, tong = 0;  
  
 printf("Nhập n: ");  
 scanf("%d", &n);  
  
 for (int i = 2; i <= n; i++) {  
 if (laSoNguyenTo(i)) tong += i;  
 }  
  
 printf("Tổng các số nguyên tố ≤ %d là: %d\n", n, tong);  
 return 0;  
}

So sánh trước và sau khi refactor

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu chí | Mã gốc | Mã refactor | Cải thiện đạt được |
| Tên biến | a, b, c, d, e, f (khó hiểu) | n, tong, i (rõ nghĩa) | Tăng tính dễ đọc, dễ hiểu |
| Cấu trúc chương trình | Tất cả trong main() | Tách riêng hàm laSoNguyenTo() | Dễ bảo trì, tái sử dụng |
| Thuật toán kiểm tra | Duyệt đến c/2 | Duyệt đến sqrt(n) | Giảm thời gian chạy |
| Chú thích | Không có | Có chú thích mô tả từng phần | Dễ học, dễ sửa |
| Đầu ra | In câu không có định dạng rõ | Có định dạng, thông báo rõ ràng | Tăng tính thân thiện với người dùng |