Mã bạn đã đăng được viết bằng C++, không phải C. Ký hiệu & trong các tham số hàm (int &a, int &b) là một tính năng của C++ được gọi là “truyền tham chiếu”. Điều này cho phép hàm sửa đổi các biến gốc được truyền vào.

Trong C, bạn có thể đạt được hiệu quả tương tự bằng cách truyền một con trỏ đến biến. Dưới đây là cách bạn có thể sửa đổi hàm của mình để hoạt động trong C:

void sapxep(int \*a, int \*b){

int tmp = \*a;

\*a = \*b;

\*b = tmp;

}

int main(){

int a, b; scanf("%d%d", &a, &b);

sapxep(&a, &b);

printf("%d %d", a, b);

return 0;

}

Trong phiên bản sửa đổi này, hàm sapxep nhận các con trỏ đến các số nguyên (int \*a, int \*b) thay vì tham chiếu. Khi gọi sapxep trong hàm main, bạn cần truyền địa chỉ của a và b bằng cách sử dụng toán tử &.

The code you’ve posted is written in C++, not C. The & symbol in the function parameters (int &a, int &b) is a feature of C++ known as “pass by reference”. This allows the function to modify the original variables passed into it.

In C, you can achieve a similar effect by passing a pointer to the variable. Here’s how you can modify your function to work in C:

void sapxep(int \*a, int \*b){

int tmp = \*a;

\*a = \*b;

\*b = tmp;

}

int main(){

int a, b; scanf("%d%d", &a, &b);

sapxep(&a, &b);

printf("%d %d", a, b);

return 0;

}

In this modified version, the sapxep function takes pointers to integers (int \*a, int \*b) instead of references. [When calling sapxep in the main function, you need to pass the addresses of a and b using the & operator1](https://en.cppreference.com/w/cpp/language/operator_logical)[2](https://en.cppreference.com/w/cpp/language/operator_precedence).