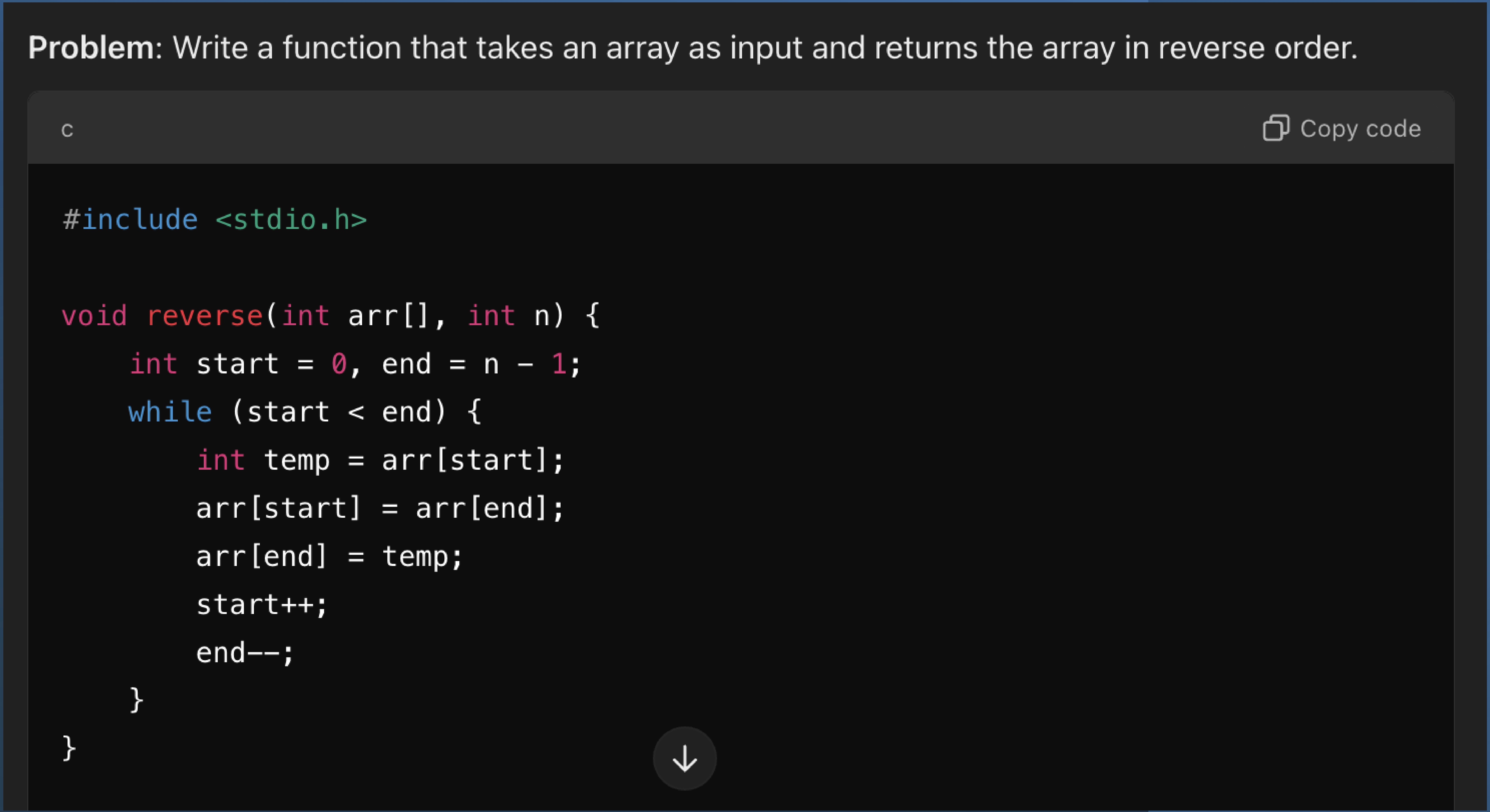
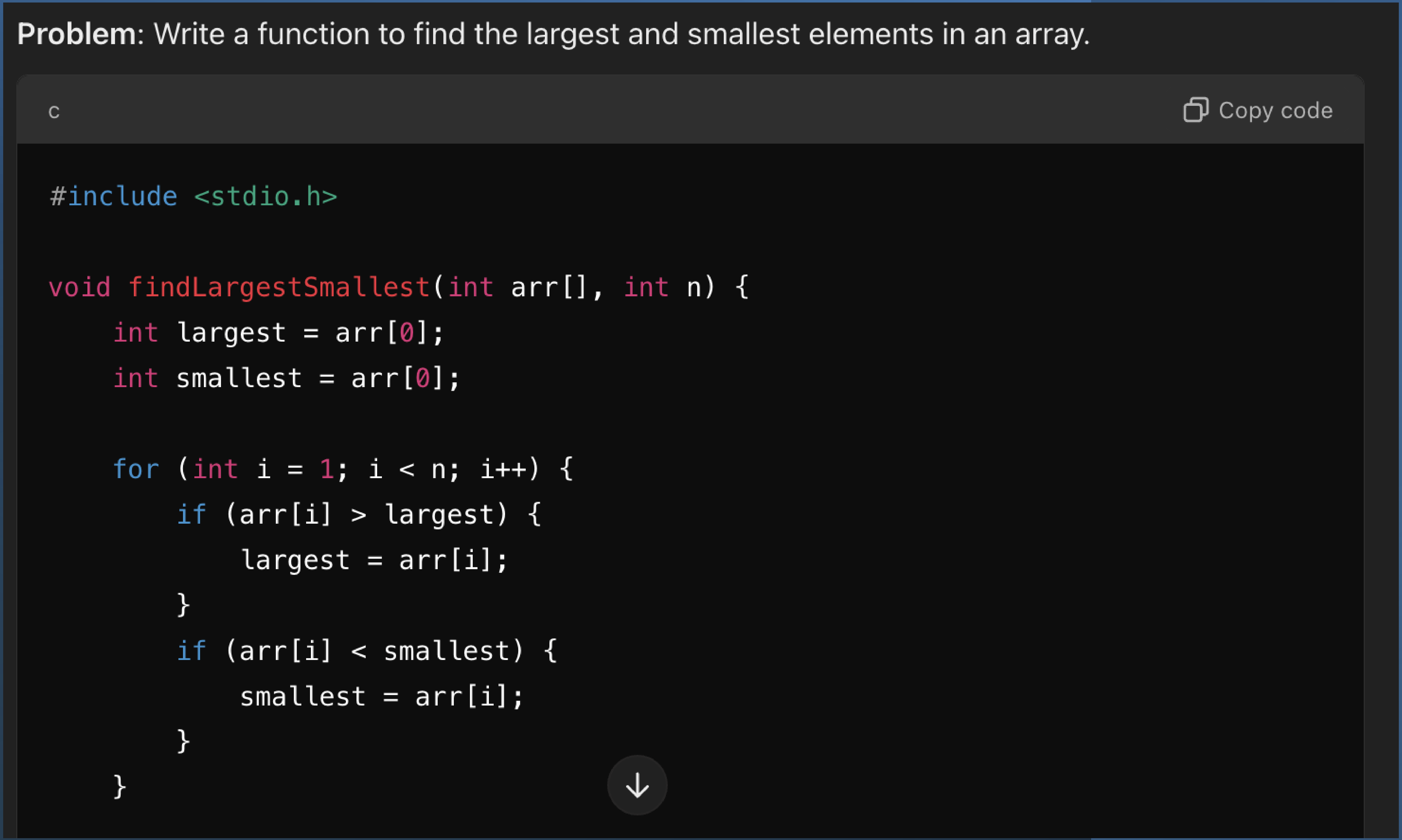
# C PROGRAMMING

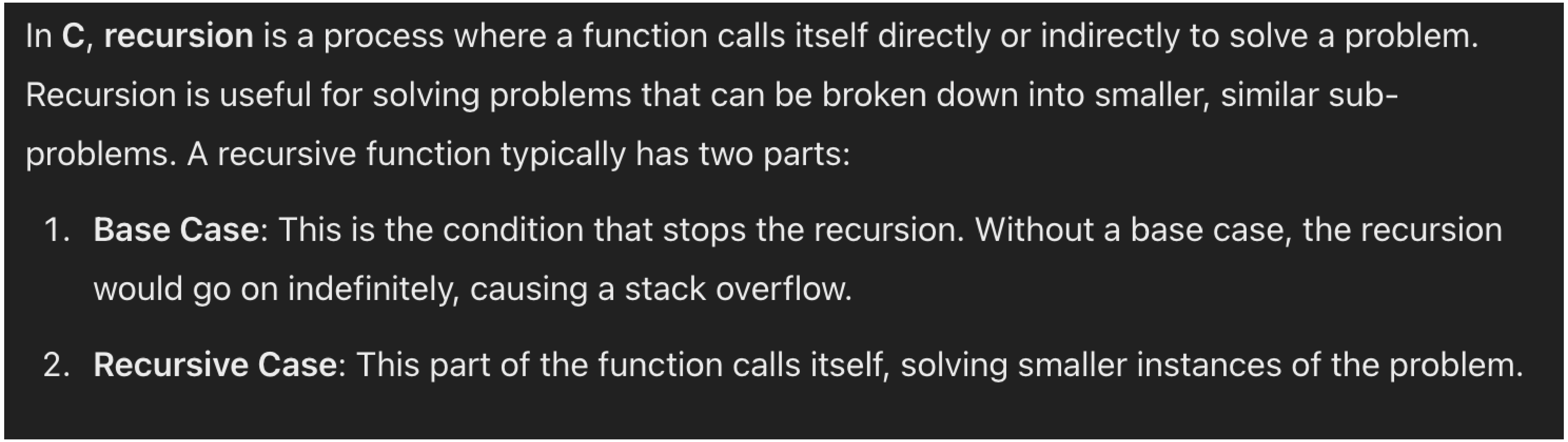
By Lungsom Lamnio


**Problem:** Write a function that takes an array as input and returns the array in reverse order.

```c
#include <stdio.h>

void reverse(int arr[], int n) {
    int start = 0, end = n - 1;
    while (start < end) {
        int temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;
        start++;
        end--;
    }
}
```

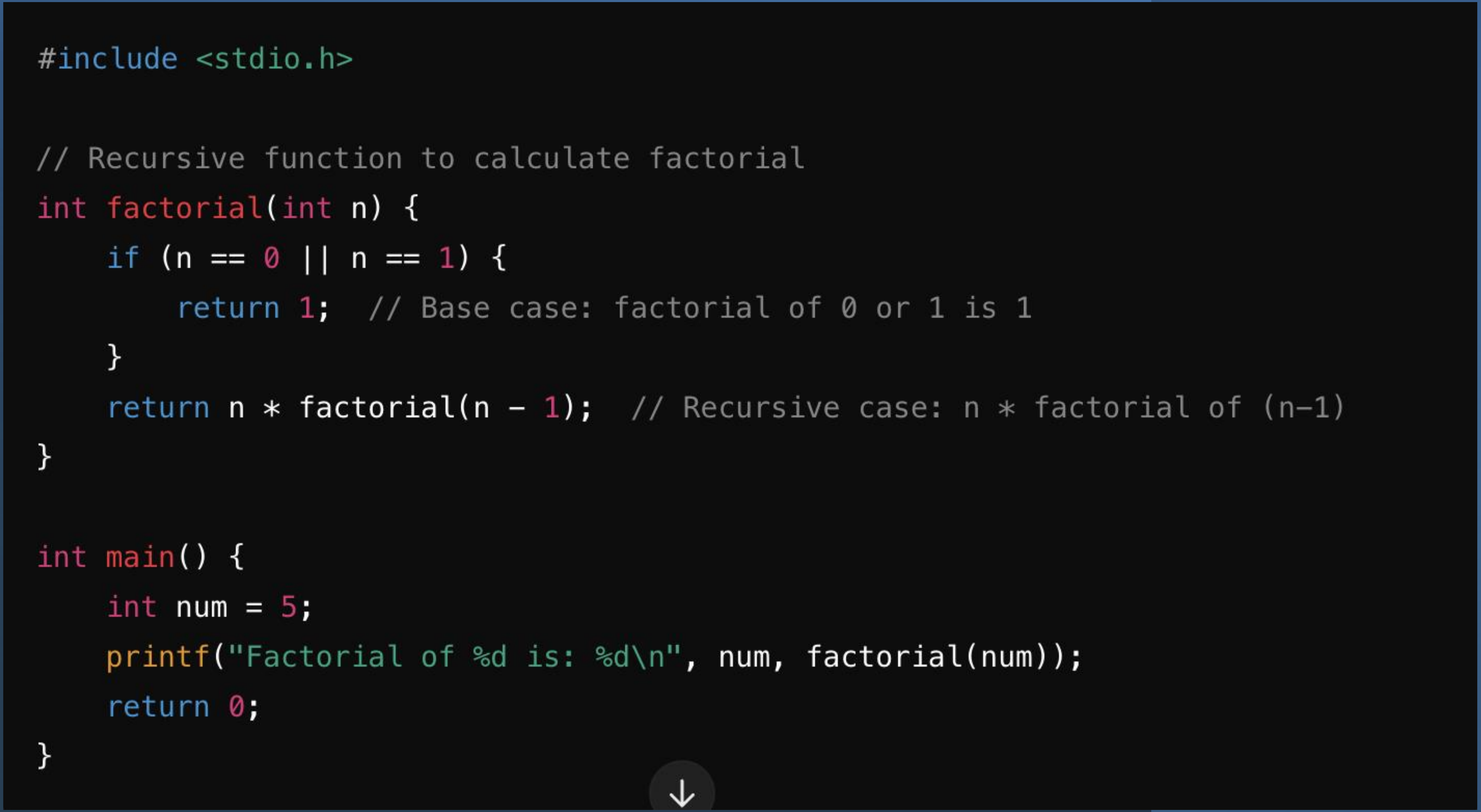

**Problem**: Write a function to find the largest and smallest elements in an array.

```c
#include <stdio.h>

void findLargestSmallest(int arr[], int n) {
    int largest = arr[0];
    int smallest = arr[0];

    for (int i = 1; i < n; i++) {
        if (arr[i] > largest) {
            largest = arr[i];
        }
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
    }
```

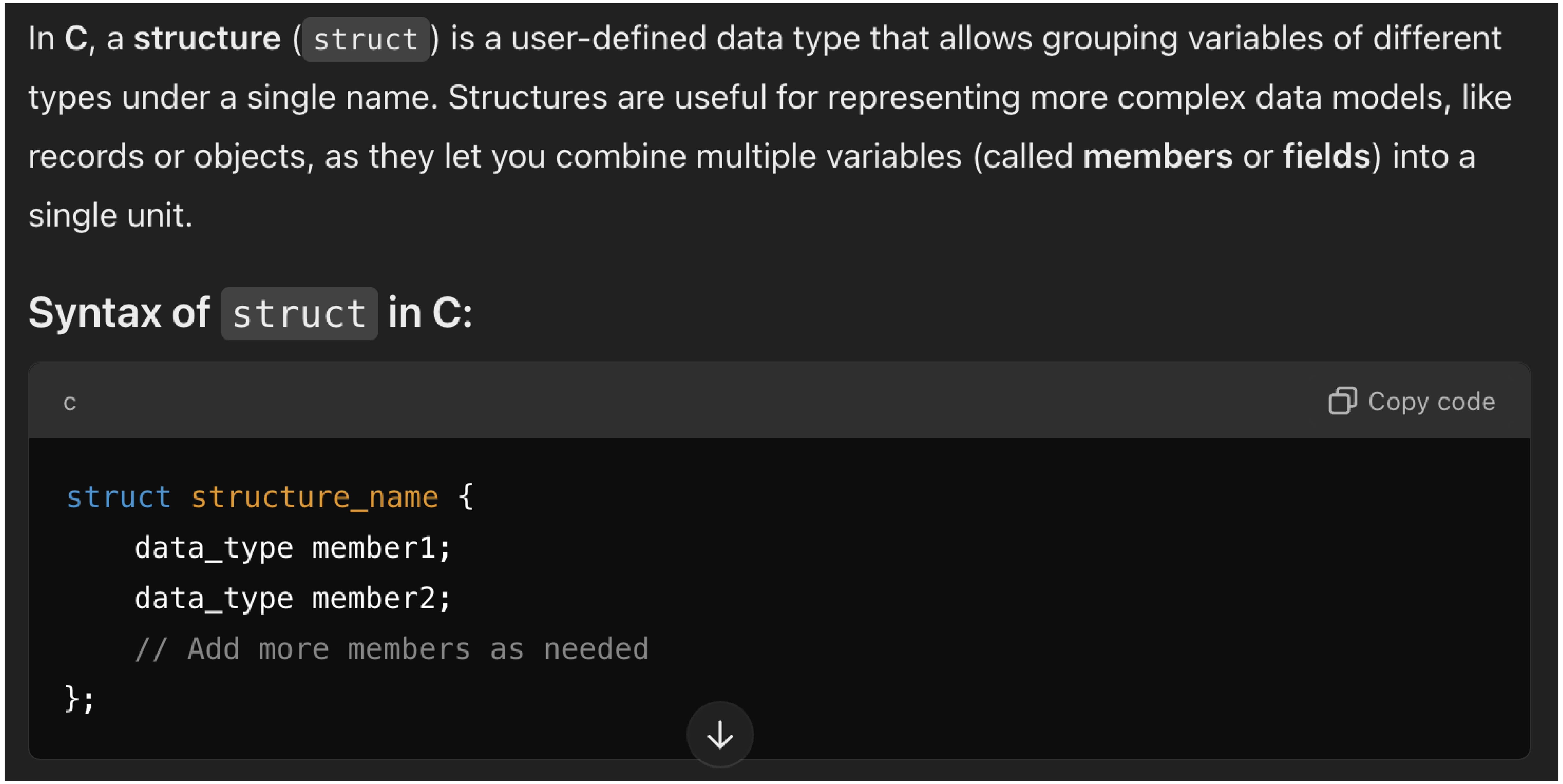


In **C, recursion** is a process where a function calls itself directly or indirectly to solve a problem. Recursion is useful for solving problems that can be broken down into smaller, similar sub-problems. A recursive function typically has two parts:

1. **Base Case**: This is the condition that stops the recursion. Without a base case, the recursion would go on indefinitely, causing a stack overflow.
2. **Recursive Case**: This part of the function calls itself, solving smaller instances of the problem.

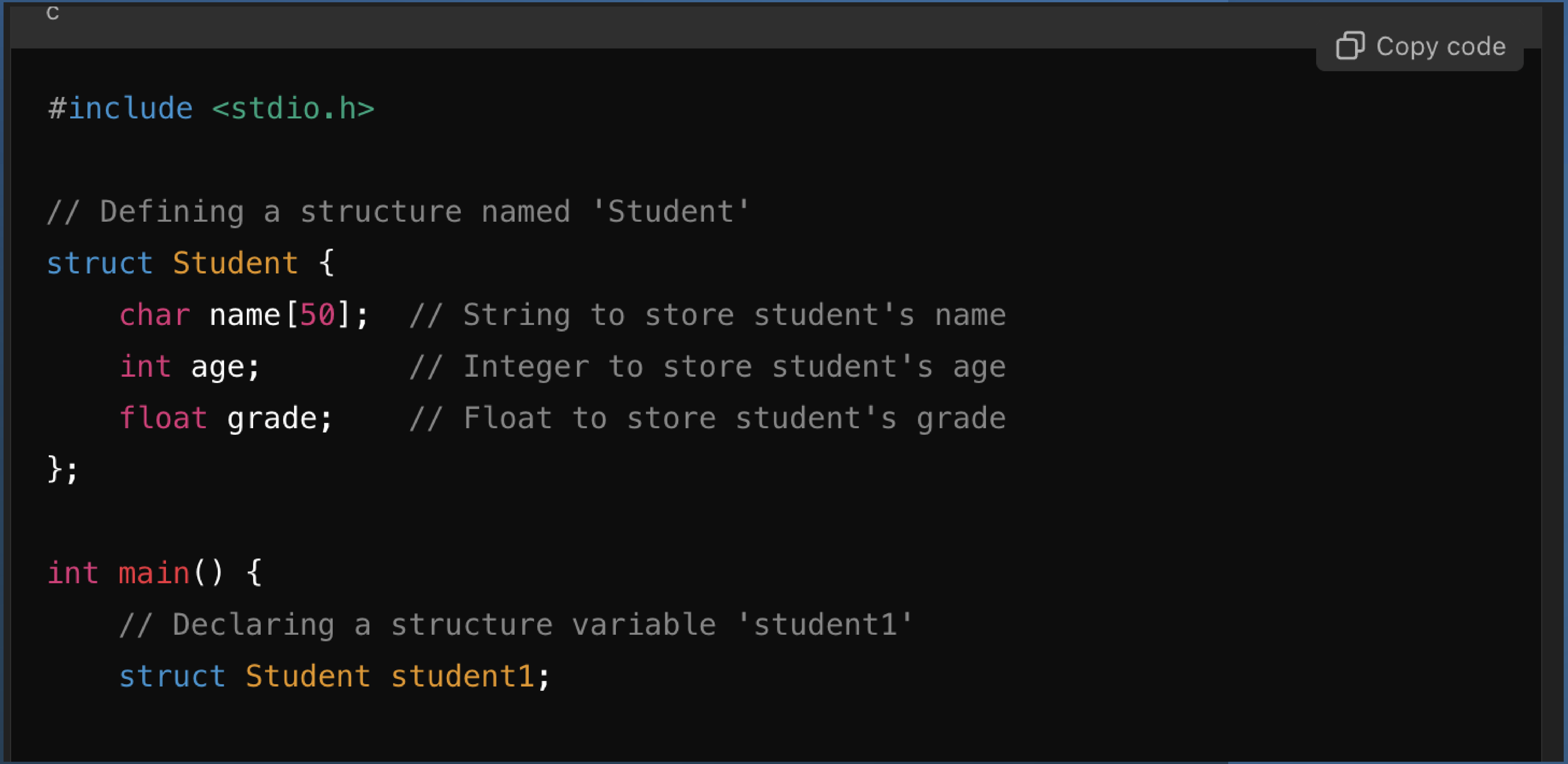


```c
#include <stdio.h>

// Recursive function to calculate factorial
int factorial(int n) {
    if (n == 0 || n == 1) {
        return 1;  // Base case: factorial of 0 or 1 is 1
    }
    return n * factorial(n - 1);  // Recursive case: n * factorial of (n-1)
}

int main() {
    int num = 5;
    printf("Factorial of %d is: %d\n", num, factorial(num));
    return 0;
}
```

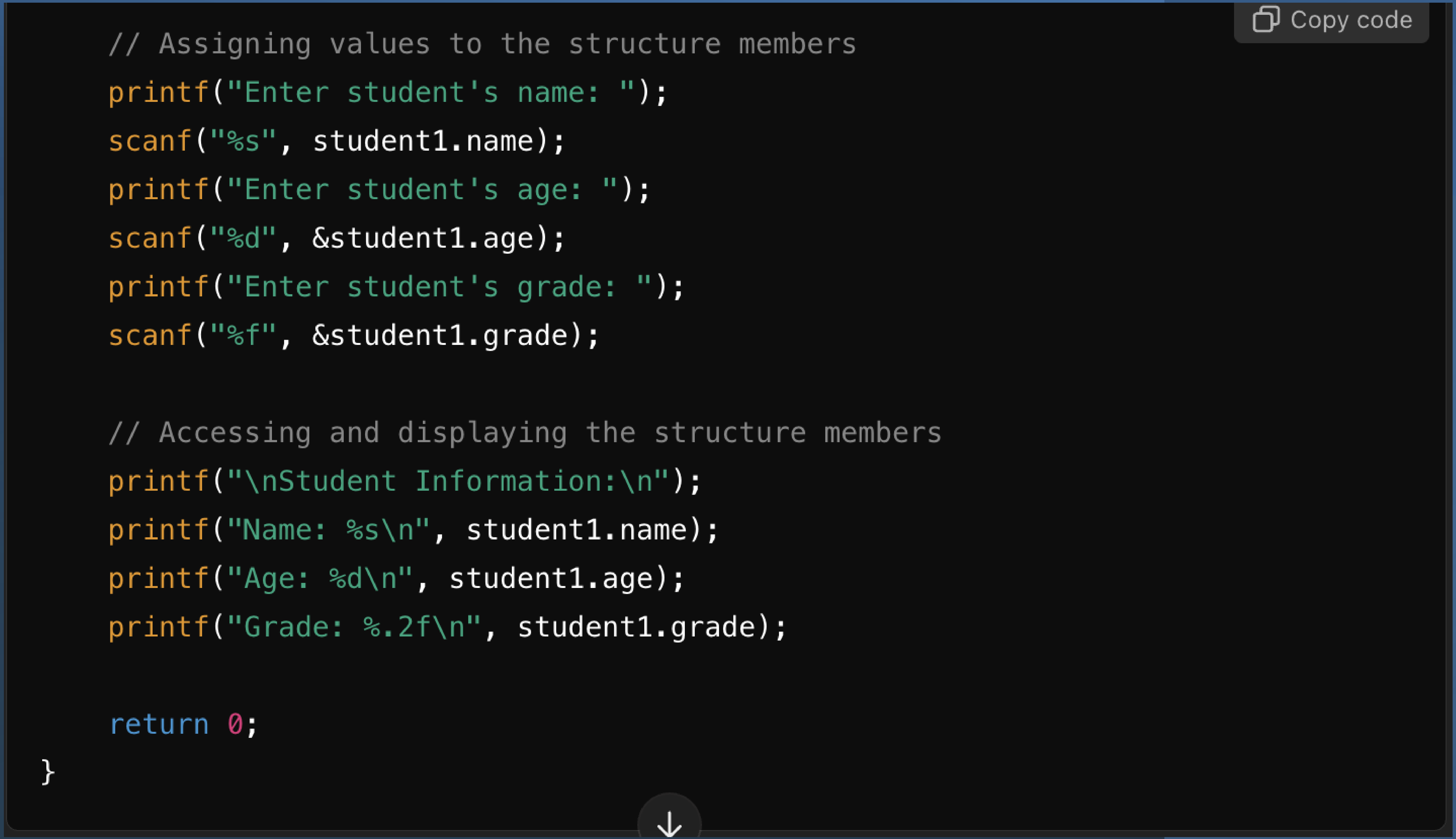

In **C**, a **structure** (`struct`) is a user-defined data type that allows grouping variables of different types under a single name. Structures are useful for representing more complex data models, like records or objects, as they let you combine multiple variables (called **members** or **fields**) into a single unit.

## Syntax of `struct` in C:

```c
struct structure_name {
    data_type member1;
    data_type member2;
    // Add more members as needed
};
```



```c
#include <stdio.h>

// Defining a structure named 'Student'
struct Student {
    char name[50];  // String to store student's name
    int age;        // Integer to store student's age
    float grade;    // Float to store student's grade
};

int main() {
    // Declaring a structure variable 'student1'
    struct Student student1;
```


Copy code

```
    // Assigning values to the structure members
    printf("Enter student's name: ");
    scanf("%s", student1.name);
    printf("Enter student's age: ");
    scanf("%d", &student1.age);
    printf("Enter student's grade: ");
    scanf("%f", &student1.grade);

    // Accessing and displaying the structure members
    printf("\nStudent Information:\n");
    printf("Name: %s\n", student1.name);
    printf("Age: %d\n", student1.age);
    printf("Grade: %.2f\n", student1.grade);

    return 0;
}
```

## **No Assignments Today**

Complete your pending assignments.

## Steps to submit the assignments

**Step 1:** Make a GitHub repo with the name as YourMentor.
**Step 2:** Inside the repo YourMentor, make a folder with the name as C_BootCamp.
**Step 3:** Submit the assignment with the day number (eg: Day1_Assignment.pdf).
**Step 4:** DM me the link of repo.

**NOTE:**

1. Active members will get the chance to make a real-life project with me.
2. Advantages with Upcoming BootCamps.

THANK YOU