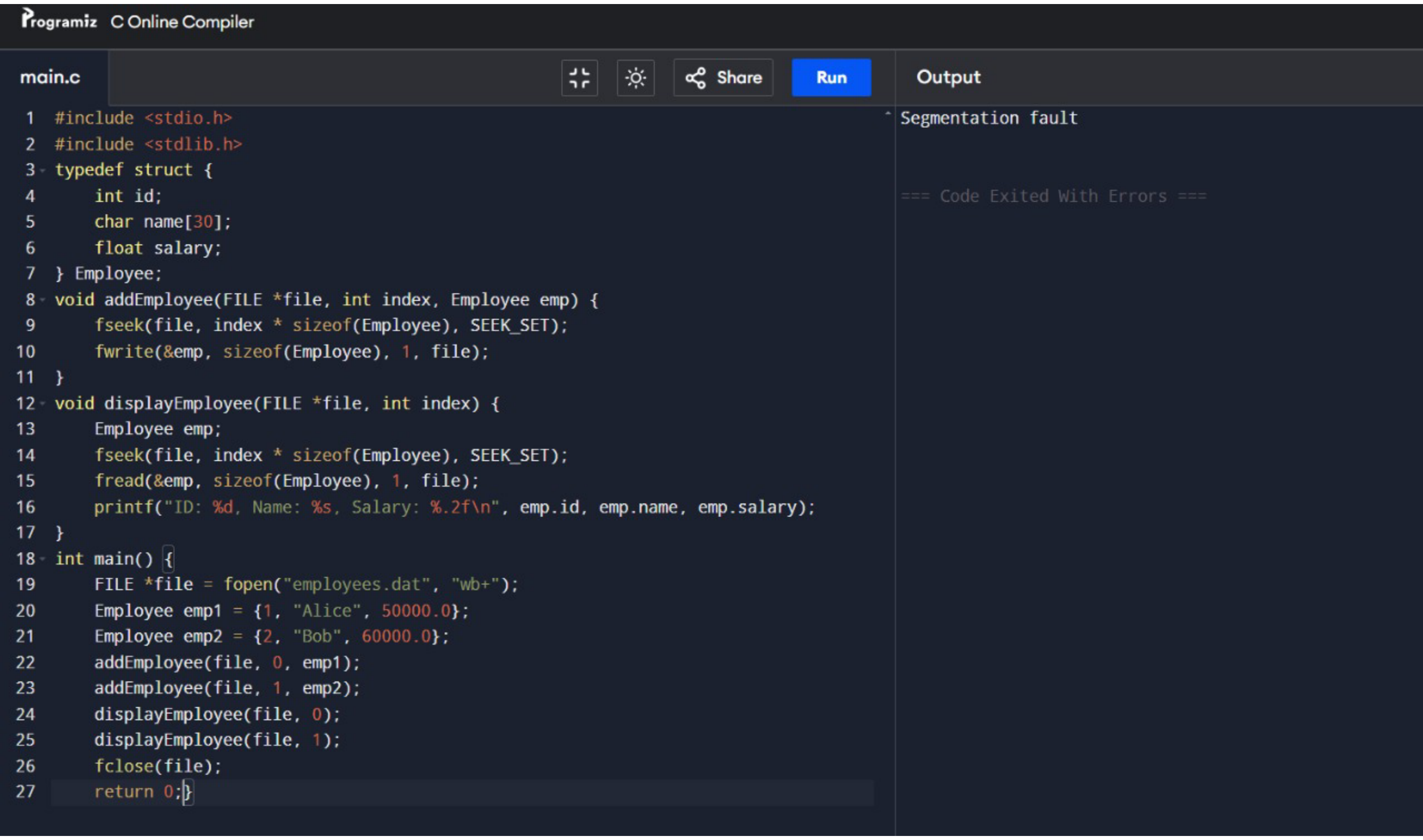
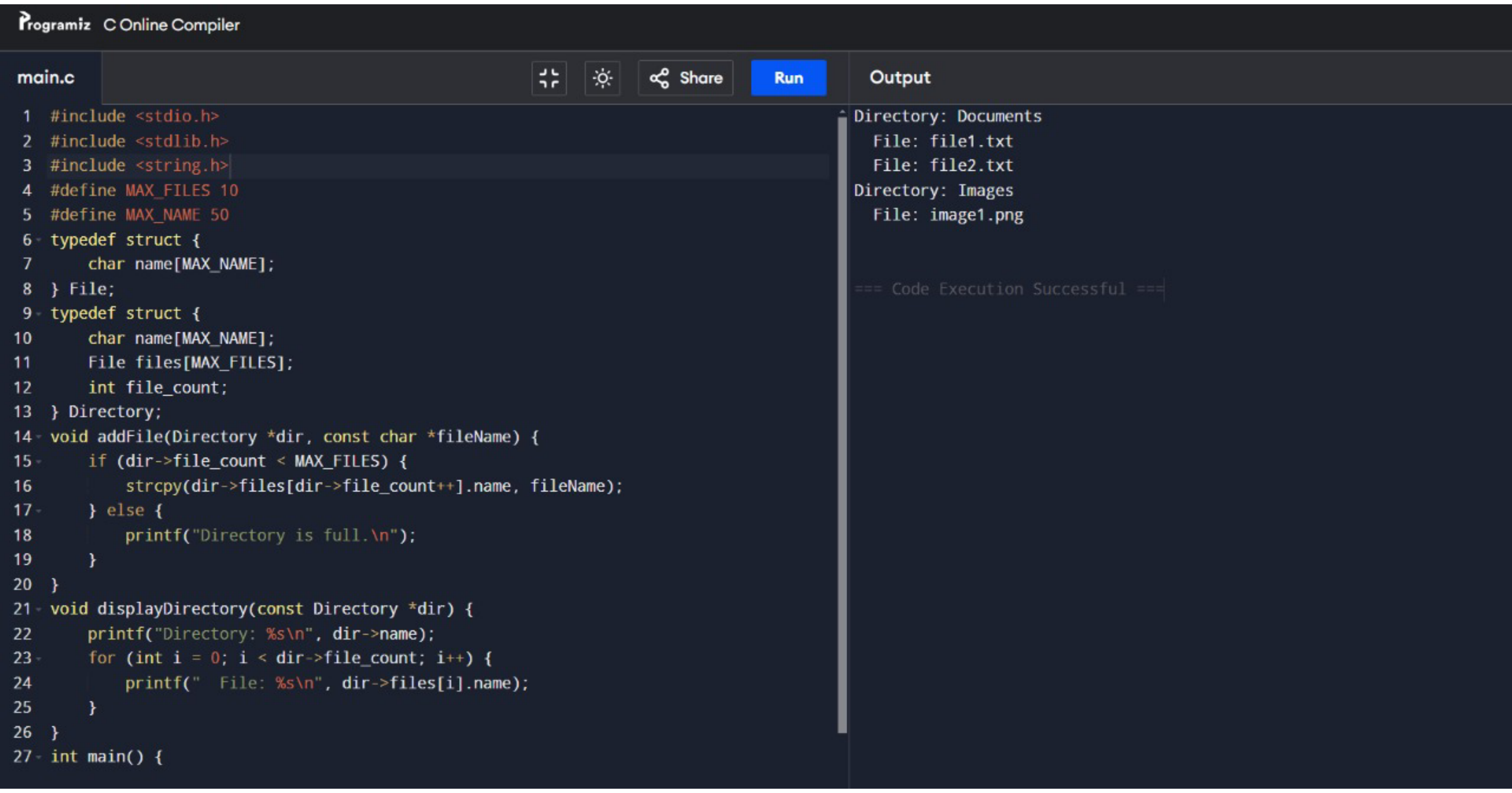


```c
#include <stdio.h>
#include <stdlib.h>
typedef struct {
    int id;
    char name[30];
    float salary;
} Employee;
void addEmployee(FILE *file, int index, Employee emp) {
    fseek(file, index * sizeof(Employee), SEEK_SET);
    fwrite(&emp, sizeof(Employee), 1, file);
}
void displayEmployee(FILE *file, int index) {
    Employee emp;
    fseek(file, index * sizeof(Employee), SEEK_SET);
    fread(&emp, sizeof(Employee), 1, file);
    printf("ID: %d, Name: %s, Salary: %.2f\n", emp.id, emp.name, emp.salary);
}
int main() {
    FILE *file = fopen("employees.dat", "wb+");
    Employee emp1 = {1, "Alice", 50000.0};
    Employee emp2 = {2, "Bob", 60000.0};
    addEmployee(file, 0, emp1);
    addEmployee(file, 1, emp2);
    displayEmployee(file, 0);
    displayEmployee(file, 1);
    fclose(file);
    return 0;}
```

```
Segmentation fault


=== Code Exited With Errors ===
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_FILES 10
#define MAX_NAME 50
typedef struct {
    char name[MAX_NAME];
} File;
typedef struct {
    char name[MAX_NAME];
    File files[MAX_FILES];
    int file_count;
} Directory;
void addFile(Directory *dir, const char *fileName) {
    if (dir->file_count < MAX_FILES) {
        strcpy(dir->files[dir->file_count++].name, fileName);
    } else {
        printf("Directory is full.\n");
    }
}
void displayDirectory(const Directory *dir) {
    printf("Directory: %s\n", dir->name);
    for (int i = 0; i < dir->file_count; i++) {
        printf("  File: %s\n", dir->files[i].name);
    }
}
int main() {
```

```
Directory: Documents
  File: file1.txt
  File: file2.txt
Directory: Images
  File: image1.png


=== Code Execution Successful ===
```