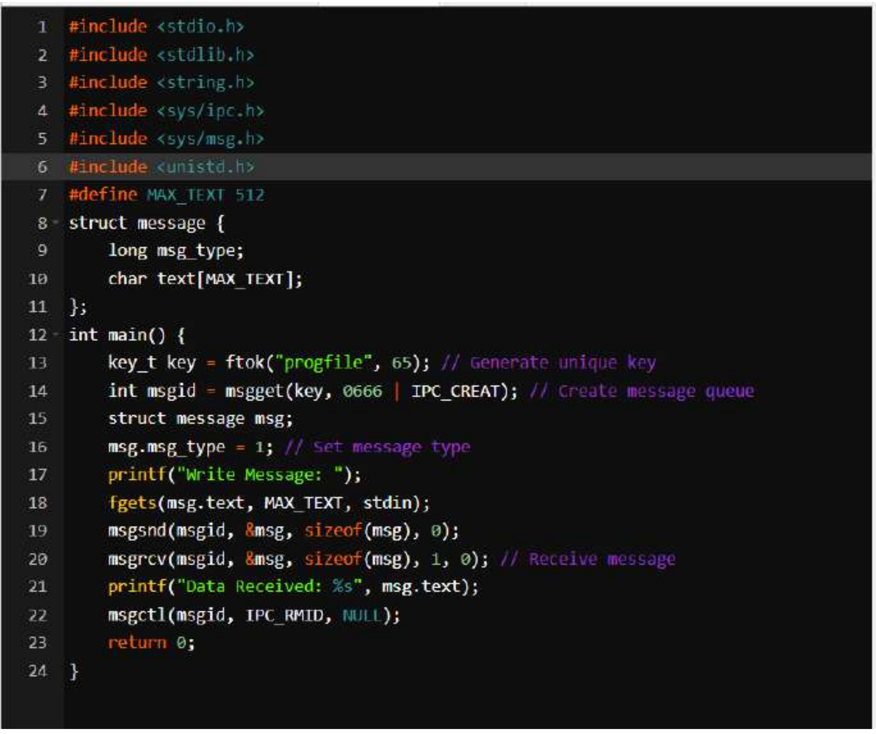
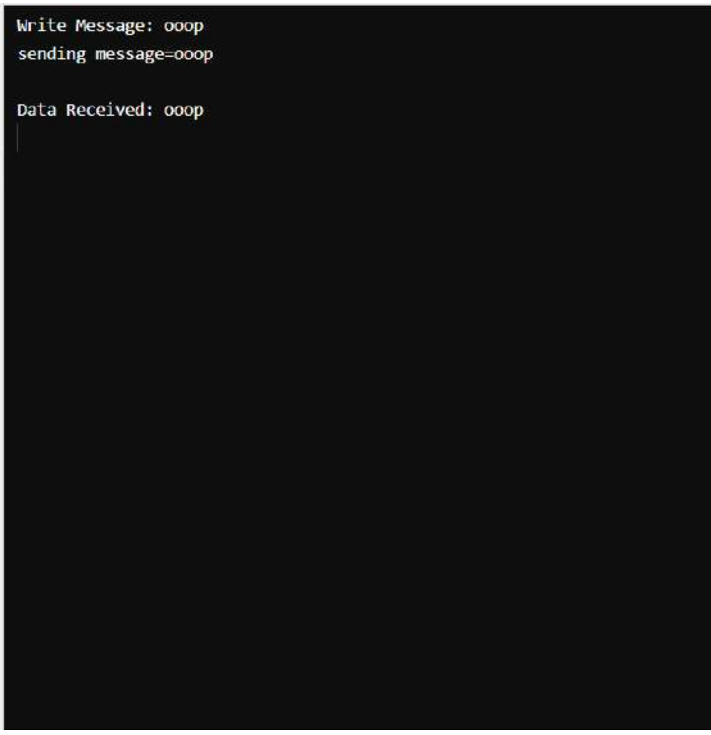
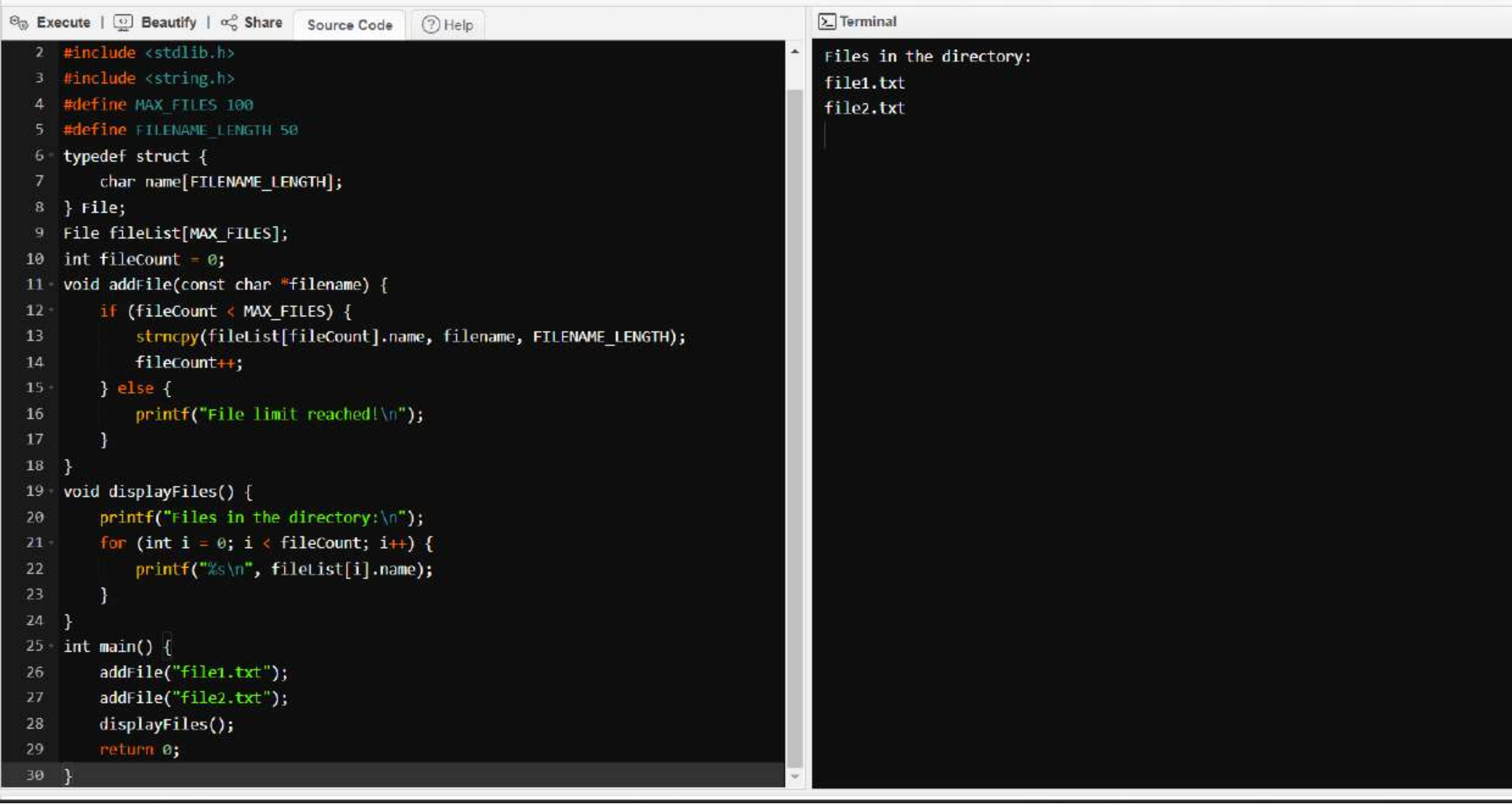


```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <unistd.h>
#define MAX_TEXT 512
struct message {
    long msg_type;
    char text[MAX_TEXT];
};
int main() {
    key_t key = ftok("progfile", 65); // Generate unique key
    int msgid = msgget(key, 0666 | IPC_CREAT); // Create message queue
    struct message msg;
    msg.msg_type = 1; // Set message type
    printf("Write Message: ");
    fgets(msg.text, MAX_TEXT, stdin);
    msgsnd(msgid, &msg, sizeof(msg), 0);
    msgrcv(msgid, &msg, sizeof(msg), 1, 0); // Receive message
    printf("Data Received: %s", msg.text);
    msgctl(msgid, IPC_RMID, NULL);
    return 0;
}
```



```
Write Message: ooop
sending message=ooop

Data Received: ooop
```



```c
#include <stdlib.h>
#include <string.h>
#define MAX_FILES 100
#define FILENAME_LENGTH 50
typedef struct {
    char name[FILENAME_LENGTH];
} File;
File fileList[MAX_FILES];
int fileCount = 0;
void addFile(const char *filename) {
    if (fileCount < MAX_FILES) {
        strncpy(fileList[fileCount].name, filename, FILENAME_LENGTH);
        fileCount++;
    } else {
        printf("File limit reached!\n");
    }
}
void displayFiles() {
    printf("Files in the directory:\n");
    for (int i = 0; i < fileCount; i++) {
        printf("%s\n", fileList[i].name);
    }
}
int main() {
    addFile("file1.txt");
    addFile("file2.txt");
    displayFiles();
    return 0;
}
```

```
Files in the directory:
file1.txt
file2.txt
```