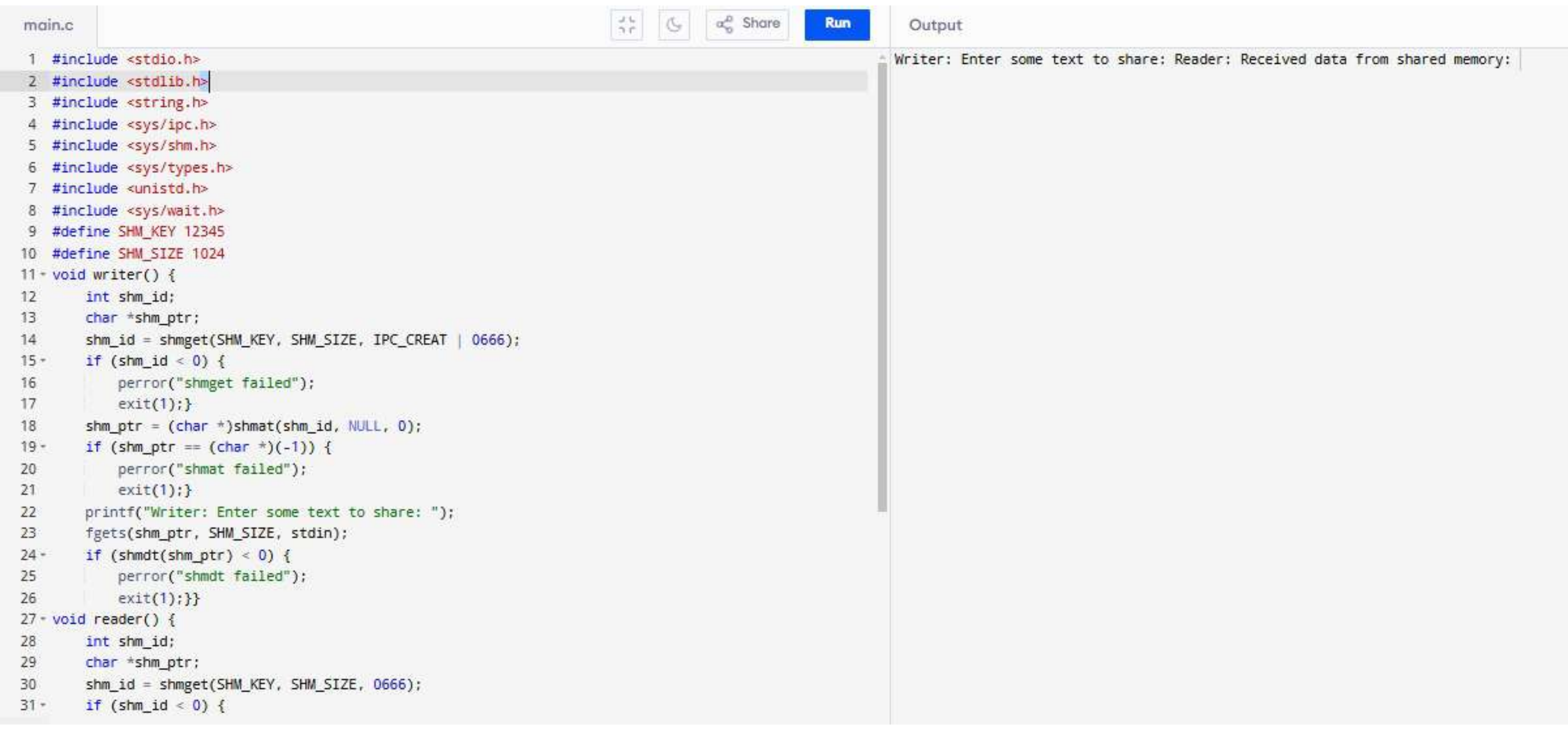
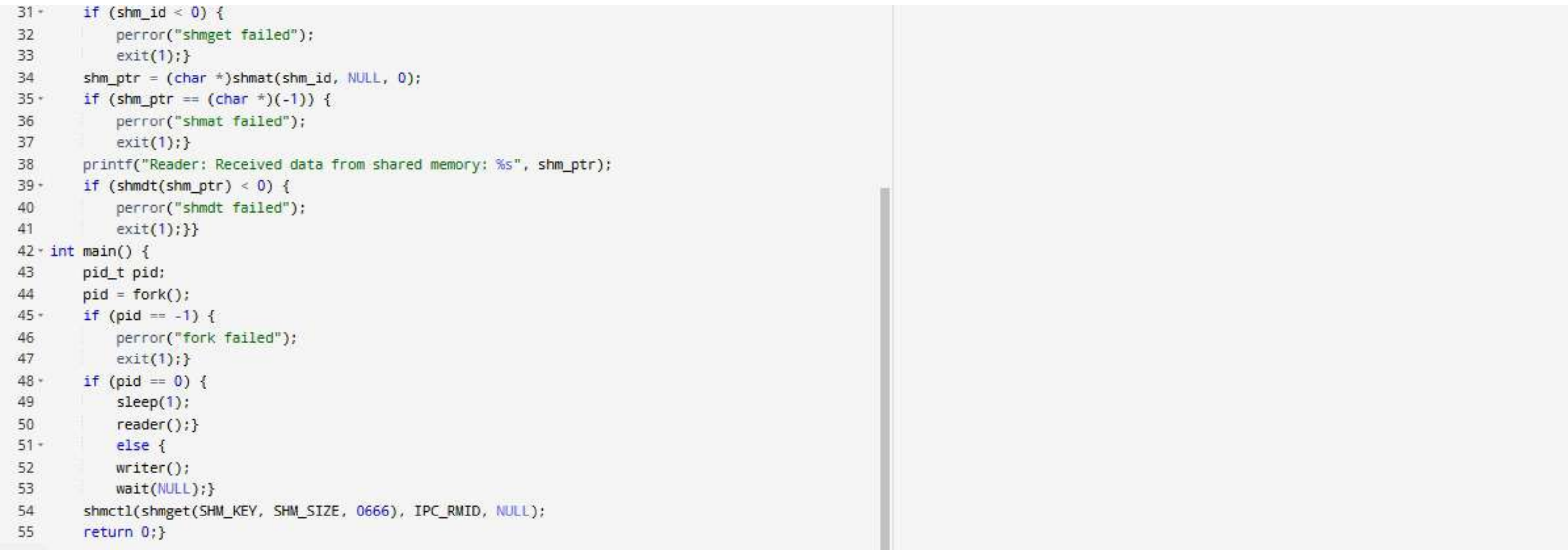
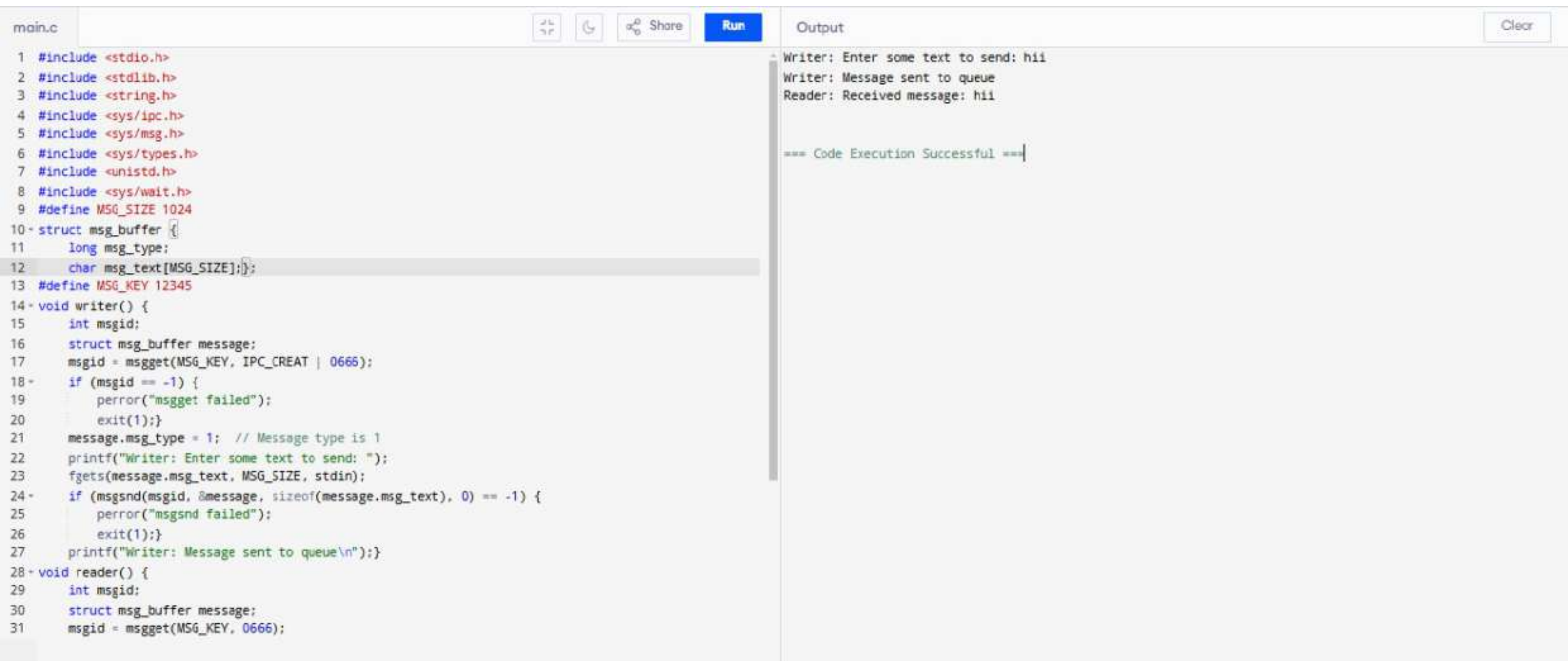


```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
#define SHM_KEY 12345
#define SHM_SIZE 1024
void writer() {
    int shm_id;
    char *shm_ptr;
    shm_id = shmget(SHM_KEY, SHM_SIZE, IPC_CREAT | 0666);
    if (shm_id < 0) {
        perror("shmget failed");
        exit(1);}
    shm_ptr = (char *)shmat(shm_id, NULL, 0);
    if (shm_ptr == (char *)(-1)) {
        perror("shmat failed");
        exit(1);}
    printf("Writer: Enter some text to share: ");
    fgets(shm_ptr, SHM_SIZE, stdin);
    if (shmdt(shm_ptr) < 0) {
        perror("shmdt failed");
        exit(1);}}
void reader() {
    int shm_id;
    char *shm_ptr;
    shm_id = shmget(SHM_KEY, SHM_SIZE, 0666);
    if (shm_id < 0) {
```

```
Writer: Enter some text to share: Reader: Received data from shared memory:
```



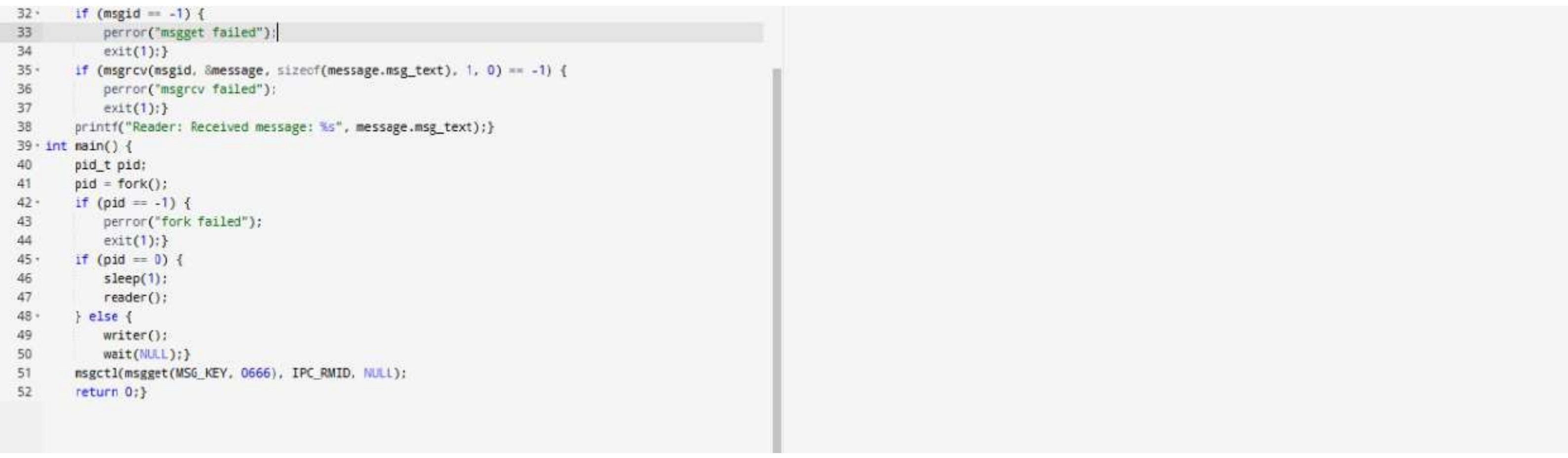
```c
    if (shm_id < 0) {
        perror("shmget failed");
        exit(1);}
    shm_ptr = (char *)shmat(shm_id, NULL, 0);
    if (shm_ptr == (char *)(-1)) {
        perror("shmat failed");
        exit(1);}
    printf("Reader: Received data from shared memory: %s", shm_ptr);
    if (shmdt(shm_ptr) < 0) {
        perror("shmdt failed");
        exit(1);}}
int main() {
    pid_t pid;
    pid = fork();
    if (pid == -1) {
        perror("fork failed");
        exit(1);}
    if (pid == 0) {
        sleep(1);
        reader();}
        else {
        writer();
        wait(NULL);}
    shmctl(shmget(SHM_KEY, SHM_SIZE, 0666), IPC_RMID, NULL);
    return 0;}
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
#define MSG_SIZE 1024
struct msg_buffer {
    long msg_type;
    char msg_text[MSG_SIZE];};
#define MSG_KEY 12345
void writer() {
    int msgid;
    struct msg_buffer message;
    msgid = msgget(MSG_KEY, IPC_CREAT | 0666);
    if (msgid == -1) {
        perror("msgget failed");
        exit(1);}
    message.msg_type = 1;  // Message type is 1
    printf("Writer: Enter some text to send: ");
    fgets(message.msg_text, MSG_SIZE, stdin);
    if (msgsnd(msgid, &message, sizeof(message.msg_text), 0) == -1) {
        perror("msgsnd failed");
        exit(1);}
    printf("Writer: Message sent to queue\n");}
void reader() {
    int msgid;
    struct msg_buffer message;
    msgid = msgget(MSG_KEY, 0666);
```

```
Writer: Enter some text to send: hii
Writer: Message sent to queue
Reader: Received message: hii


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



```
    if (msgid == -1) {
        perror("msgget failed");
        exit(1);}
    if (msgrcv(msgid, &message, sizeof(message.msg_text), 1, 0) == -1) {
        perror("msgrcv failed");
        exit(1);}
    printf("Reader: Received message: %s", message.msg_text);}
int main() {
    pid_t pid;
    pid = fork();
    if (pid == -1) {
        perror("fork failed");
        exit(1);}
    if (pid == 0) {
        sleep(1);
        reader();
    } else {
        writer();
        wait(NULL);}
    msgctl(msgget(MSG_KEY, 0666), IPC_RMID, NULL);
    return 0;}
```