Sender Code

```
#include<stdio.h>

#include<sys/types.h>

#include<sys/shm.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>


int main() {

    printf("\nI Am The Sender\n");


    void *shm;

    int shmid;

    key_t key = 1234;


    // Create shared memory

    shmid = shmget(key, 1024, IPC_CREAT | 0666);

    if (shmid == -1) {

        perror("shmget");

        exit(1);

    }

    printf("Shared Memory ID: %d\n", shmid);


    // Attach shared memory

    shm = shmat(shmid, NULL, 0);

    if (shm == (void *)-1) {
```

```
    perror("shmat");
    exit(1);
}
printf("Attached To Process At: %p\n", shm);

char message[100];
char *shared_mem = (char *)shm;
shared_mem[0] = 'S'; // 'S' means Sender's turn to send

while (1) {
    // Wait for Sender's turn ('S')
    while (shared_mem[0] != 'S') {
        sleep(1);
    }

    // Input message
    printf("Sender: Enter your message: ");
    fgets(message, sizeof(message), stdin);
    strcpy(shared_mem + 1, message); // Write message starting from index 1
    shared_mem[0] = 'R';            // Set flag to 'R' (Receiver's turn)

    // Exit if the message is "exit"
    if (strncmp(message, "exit", 4) == 0) {
        break;
    }
}
```

```
    return 0;

}
```

Receiver Code

```
#include<stdio.h>

#include<sys/types.h>

#include<sys/shm.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>


int main() {

    printf("\nI Am The Receiver\n");


    void *shm;

    int shmid;

    key_t key = 1234;


    // Get shared memory

    shmid = shmget(key, 1024, 0666);

    if (shmid == -1) {

        perror("shmget");

        exit(1);

    }

    printf("Shared Memory ID: %d\n", shmid);


    // Attach shared memory

    shm = shmat(shmid, NULL, 0);

    if (shm == (void *)-1) {
        perror("shmat");
        exit(1);
    }
    printf("Attached To Process At: %p\n", shm);

    char *shared_mem = (char *)shm;

    char message[100];
    while (1) {
        // Wait for Receiver's turn ('R')
        while (shared_mem[0] != 'R') {
            sleep(1);
        }

        // Display message from Sender
        printf("Receiver: Received message: %s", shared_mem + 1);

        // Exit if the message is "exit"
        if (strncmp(shared_mem + 1, "exit", 4) == 0) {
            break;
        }

        // Input reply
        printf("Receiver: Enter your reply: ");
        fgets(message, sizeof(message), stdin);
        strcpy(shared_mem + 1, message); // Write reply starting from index 1
        shared_mem[0] = 'S';            // Set flag to 'S' (Sender's turn)
        // Exit if the reply is "exit"
        if (strncmp(message, "exit", 4) == 0) {
            break;
        }
    }

    return 0;
}
```