

7.

CODE-

```
cse16@localhost:~  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/ipc.h>  
#include <sys/shm.h>  
#include <unistd.h>  
  
#define SHM_SIZE 1024 // Size of the shared memory segment  
  
int main() {  
    key_t key = 1234; // Unique key for the shared memory segment  
    int shmid;  
    char *shmaddr;  
    if ((shmid = shmget(key, SHM_SIZE, IPC_CREAT | 0666)) < 0) {  
        perror("shmget");  
        exit(1);  
    }  
    if ((shmaddr = (char *)shmat(shmid, NULL, 0)) == (char *)-1) {  
        perror("shmat");  
        exit(1);  
    }  
    char message[] = "Hello from the sender process!";  
    sprintf(shmaddr, "%s", message);  
    printf("Sender wrote: %s\n", message);  
  
    sleep(5);  
  
    if (shmdt(shmaddr) == -1) {  
        perror("shmdt");  
        exit(1);  
    }  
  
    return 0;  
}
```

```
cse16@localhost:~  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/ipc.h>  
#include <sys/shm.h>  
#include <unistd.h>  
  
#define SHM_SIZE 1024 // Size of the shared memory segment  
  
int main() {  
    key_t key = 1234; // Unique key for the shared memory segment  
    int shmid;  
    char *shmaddr;  
  
    // 1. Allocate the shared memory segment (it should already exist)  
    if ((shmid = shmget(key, SHM_SIZE, 0666)) < 0) {  
        perror("shmget");  
        exit(1);  
    }  
  
    // 2. Attach the shared memory segment  
    if ((shmaddr = (char *)shmat(shmid, NULL, 0)) == (char *)-1) {  
        perror("shmat");  
        exit(1);  
    }  
  
    // 3. Print the shared memory contents  
    printf("Message Received: %s\n", shmaddr);  
  
    // 4. Detach shared memory segment  
    if (shmdt(shmaddr) == -1) {  
        perror("shmdt");  
        exit(1);  
    }  
  
    // To remove the shared memory segment (optional):  
    // if (shmctl(shmid, IPC_RMID, NULL) == -1) {  
    //     perror("shmctl");  
    //     exit(1);  
    // }  
  
    return 0;  
}
```

OUTPUT-

```
cse16@localhost:~  
login as: cse16  
cse16@172.16.8.127's password:  
Last login: Mon Apr  7 08:10:15 2025 from 172.16.9.23  
[cse16@localhost ~]$ vi sender.c  
[cse16@localhost ~]$ vi receiver.c  
[cse16@localhost ~]$ gcc sender.c -o sender  
[cse16@localhost ~]$ gcc receiver.c -o receiver  
[cse16@localhost ~]$ ./sender  
Sender wrote: Hello from the sender process!  
[cse16@localhost ~]$
```

```
cse16@localhost:~  
login as: cse16  
cse16@172.16.8.127's password:  
Last login: Mon Apr  7 08:16:38 2025 from 172.16.9.23  
[cse16@localhost ~]$ ./receiver  
Message Received: Hello from the sender process!  
[cse16@localhost ~]$
```