**OS Lab**

**Session 3 – Lab 6 (15/12/2020)**

Parthivi Choubey CSE – B - 5th semester

180905456 Roll. no. - 60

**Question 1**

**Code**

Sender

Receiver

**Output**

**Question 2**

**Code**

#include <stdlib.h>

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <sys/shm.h>

#include <sys/wait.h>

#define SIZE 2

void getNextAlphabet (char \*a)

{

char \*n = (char \*)calloc(2, sizeof(char));

n[1] = '\0';

if (\*a == 'Z')

n[0] = 'a';

else if (\*a == 'z')

n[0] = 'A';

else

n[0] = \*a + 1;

\*a = \*n;

}

int main (int argc, const char \* argv [])

{

int shmid = shmget(IPC\_PRIVATE, SIZE, S\_IRUSR | S\_IWUSR);

if(shmid==-1)

{

fprintf(stderr, "shmget failed\n");

exit(EXIT\_FAILURE);

}

char \*shared\_memory = (char \*)shmat(shmid, NULL, 0);

if(shared\_memory==(void\*)-1)

{

fprintf(stderr, "shmat failed\n");

exit(EXIT\_FAILURE);

}

\*shared\_memory = '\0';

printf("Enter an alphabet: ");

scanf("%c", shared\_memory);

pid\_t pid = fork();

if (pid == 0) //child process

{

while (\*shared\_memory == '\0');

getNextAlphabet(shared\_memory);

exit(0);

}

else //parent process

{

printf("%s -> ", shared\_memory);

wait(NULL);

printf("%s\n", shared\_memory);

}

if(shmdt(shared\_memory)==-1)

{

fprintf(stderr, "shmdt failed\n");

exit(EXIT\_FAILURE);

}

if(shmctl(shmid, IPC\_RMID, NULL)==-1)

{

fprintf(stderr, "IPC\_RMID failed\n");

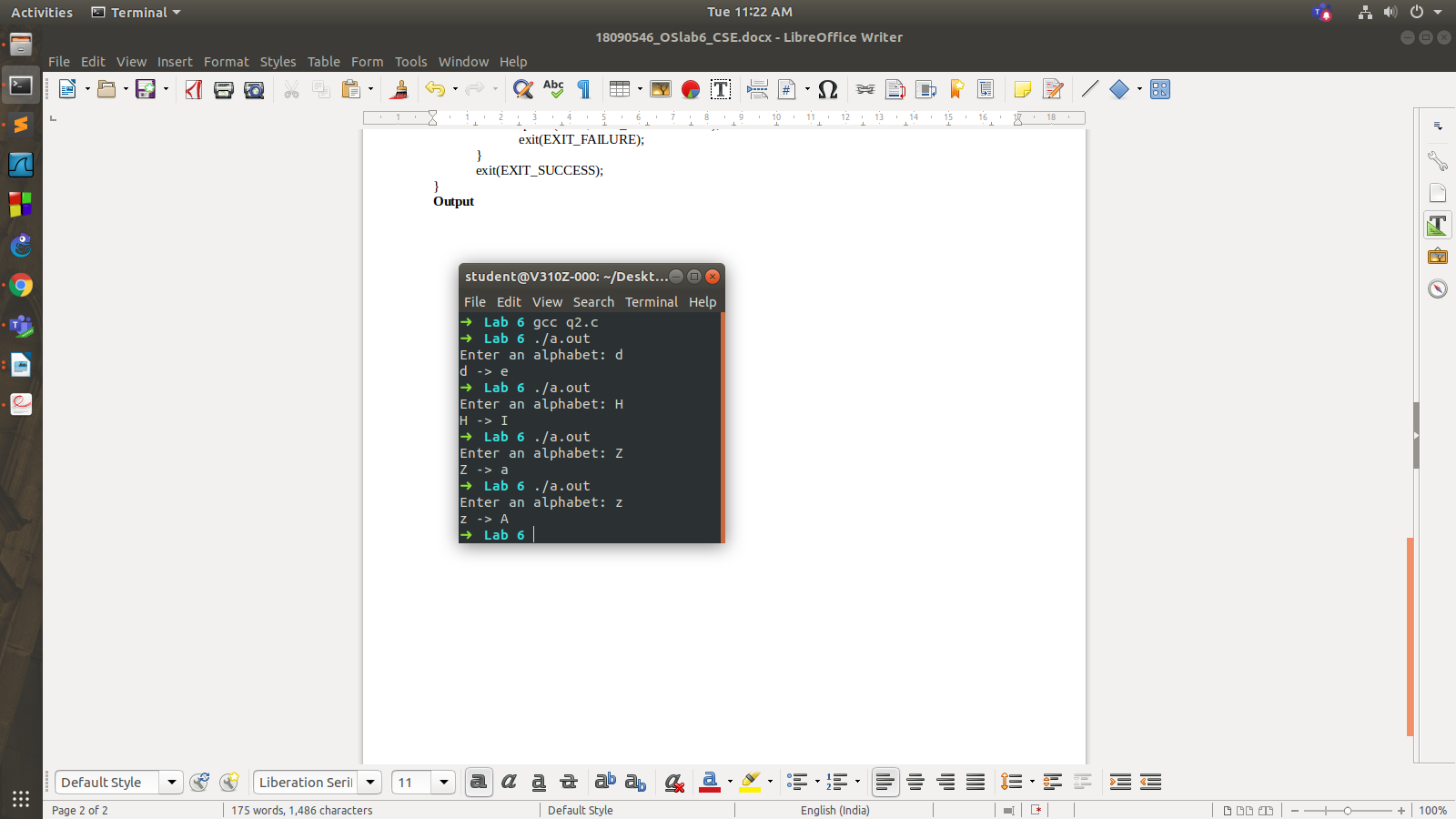
exit(EXIT\_FAILURE);

}

exit(EXIT\_SUCCESS);

}

**Output**

****