**19CSE213 OS Periodical Lab Test**

**CSE-E**

**Question :**

Consider a system with 2 types of processes. The billing process keeps track of product details such as product-id, name, quantity, and unit price. Billing process starts up by initializing stock and other parameters. Customer process submits the item and quantity to the billing process. Billing process computes the total amount including 18% GST for the items.

The customer processes use message queue to communicate to the billing process.

**Source Code :**

**Receiver :**

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 128

void die(char \*s)

{

perror(s);

exit(1);

}

struct mesg\_buffer

{

long mtype;

char mtext[MAXSIZE];

}rcv ;

int main()

{

int msqid;

key\_t key\_v1,key\_v2,key\_v3,key\_v4,key\_v5;

int c1=0,c2=0,c3=0,winner;

key\_v1 = 1234;

key\_v2 = 12345; key\_v3 = 123456;key\_v4=1234567;key\_v5=1234577;

if ((msqid = msgget(key\_v1, 0666)) < 0)

die("msgget()");

if (msgrcv(msqid, &rcv, sizeof(rcv), 1, 0) < 0)

die("msgrcv");

int amount=0;

char \*s=rcv.mtext[0];

int id=atoi(s);

int qty=0;

int k;

for(int i=1;rcv.mtext[i]!='\0';i++)

{

\*s=rcv.mtext[i];

k=atoi(s);

qty=qty\*10+k;

}

if(id==1)

{

amount=amount+qty\*10;

}

if(id==2)

{

amount=amount+qty\*15;

}

if(id==3)

{

amount=amount+qty\*20;

}

float fam=amount\*1.18;

printf("customer1:%f\n",fam);

if ((msqid = msgget(key\_v2, 0666)) < 0)

die("msgget()");

if (msgrcv(msqid, &rcv, MAXSIZE, 1, 0) < 0)

die("msgrcv");

amount=0;

\*s=rcv.mtext[0];

id=atoi(s);

qty=0;

for(int i=1;rcv.mtext[i]!='\0';i++)

{

\*s=rcv.mtext[i];

k=atoi(s);

qty=qty\*10+k;

}

if(id==1)

{

amount=amount+qty\*10;

}

if(id==2)

{

amount=amount+qty\*15;

}

if(id==3)

{

amount=amount+qty\*20;

}

fam=amount\*1.18;

printf("customer2:%f\n",fam);

if ((msqid = msgget(key\_v3, 0666)) < 0)

die("msgget()");

if (msgrcv(msqid, &rcv, MAXSIZE, 1, 0) < 0)

die("msgrcv");

amount=0;

\*s=rcv.mtext[0];

id=atoi(s);

qty=0;

for(int i=1;rcv.mtext[i]!='\0';i++)

{

\*s=rcv.mtext[i];

k=atoi(s);

qty=qty\*10+k;

}

if(id==1)

{

amount=amount+qty\*10;

}

if(id==2)

{

amount=amount+qty\*15;

}

if(id==3)

{

amount=amount+qty\*20;

}

fam=amount\*1.18;

printf("customer3:%f\n",fam);

if ((msqid = msgget(key\_v4, 0666)) < 0)

die("msgget()");

if (msgrcv(msqid, &rcv, MAXSIZE, 1, 0) < 0)

die("msgrcv");

amount=0;

\*s=rcv.mtext[0];

id=atoi(s);

qty=0;

for(int i=1;rcv.mtext[i]!='\0';i++)

{

\*s=rcv.mtext[i];

k=atoi(s);

qty=qty\*10+k;

}

if(id==1)

{

amount=amount+qty\*10;

}

if(id==2)

{

amount=amount+qty\*15;

}

if(id==3)

{

amount=amount+qty\*20;

}

fam=amount\*1.18;

printf("customer4:%f\n",fam);

if ((msqid = msgget(key\_v5, 0666)) < 0)

die("msgget()");

if (msgrcv(msqid, &rcv, MAXSIZE, 1, 0) < 0)

die("msgrcv");

amount=0;

\*s=rcv.mtext[0];

id=atoi(s);

qty=0;

for(int i=1;rcv.mtext[i]!='\0';i++)

{

\*s=rcv.mtext[i];

k=atoi(s);

qty=qty\*10+k;

}

if(id==1)

{

amount=amount+qty\*10;

}

if(id==2)

{

amount=amount+qty\*15;

}

if(id==3)

{

amount=amount+qty\*20;

}

fam=amount\*1.18;

printf("customer5:%f\n",fam);

return 0;

}

**Sender Code :**

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAXSIZE 128

typedef struct msgbuf1

{

long mtype;

char mtext[MAXSIZE];

}msgbuf;

void die(char \*s)

{

perror(s);

exit(1);

}

int main()

{

int msqid;

int msgflg = IPC\_CREAT | 0666;

key\_t key;

msgbuf sbuf;

size\_t buflen;

key = 1234;

if ((msqid = msgget(key, msgflg )) < 0)

die("msgget");

sbuf.mtype = 1;

printf("Welcome customer 1 \n");

printf("Please enter item id, product1-1 product-2 product-3\n");

scanf("%[^\n]",sbuf.mtext);

getchar();

if (msgsnd(msqid, &sbuf, sizeof(sbuf), 0) < 0)

{

printf ("not sent\n");

die("msgsnd");

}

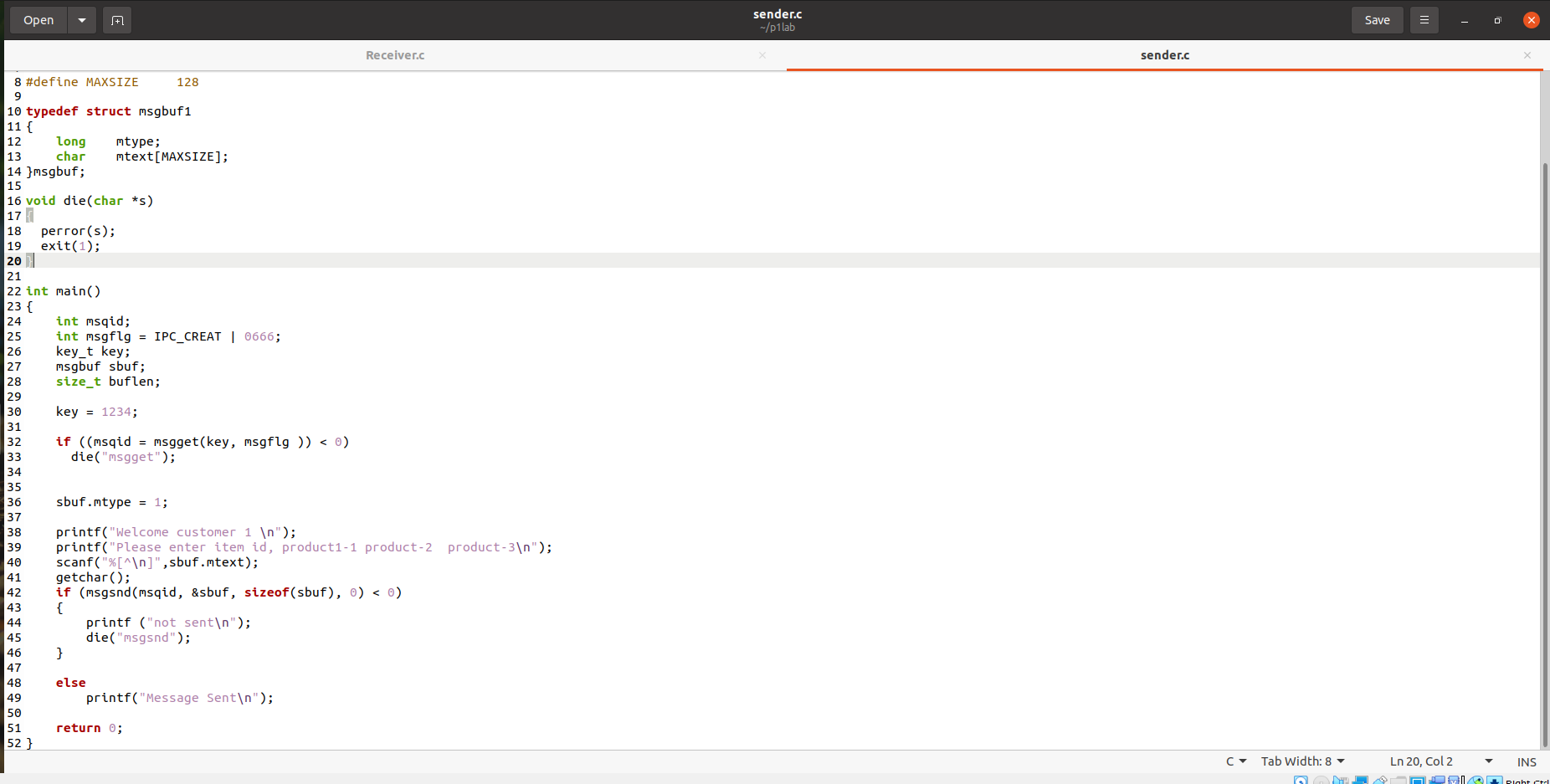
else

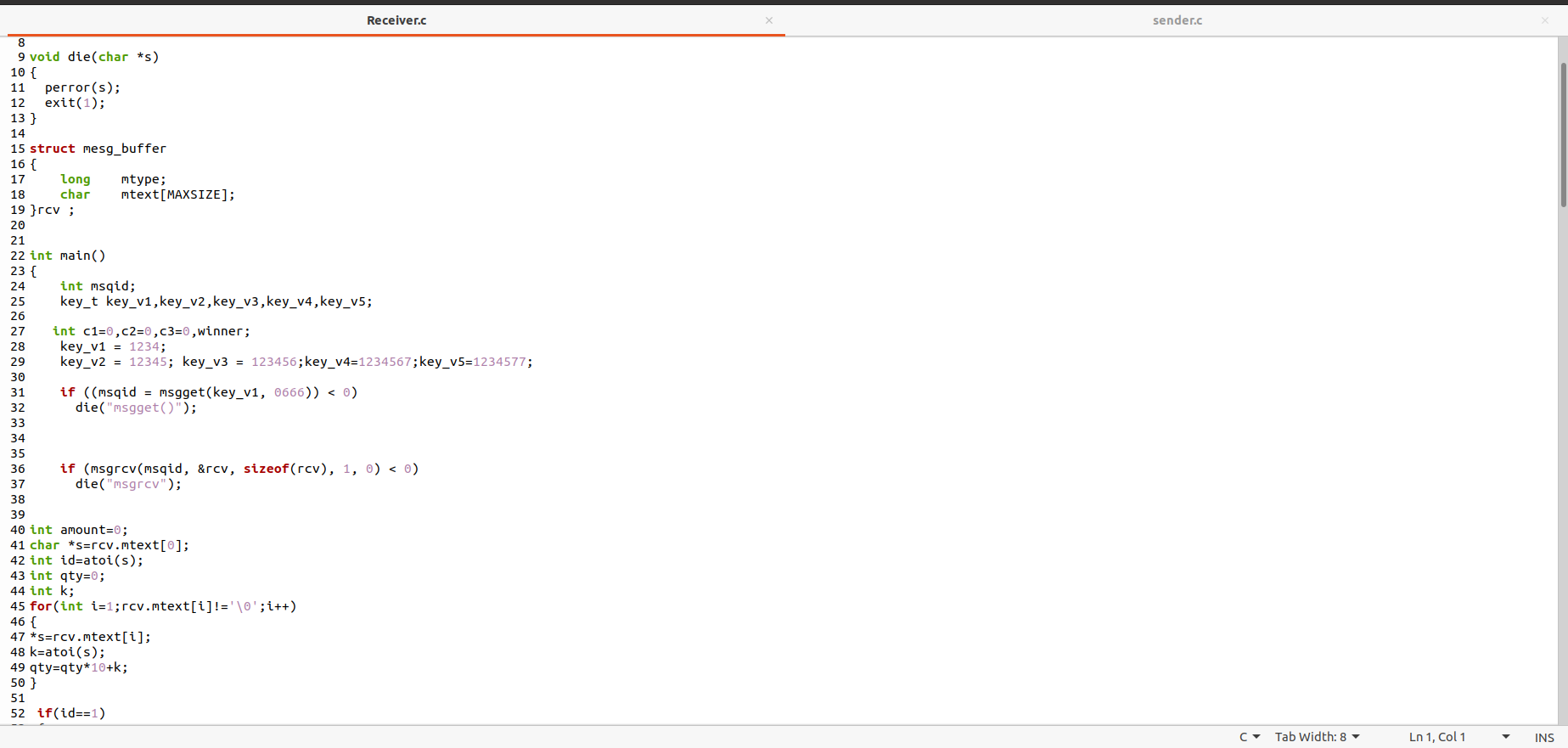
printf("Message Sent\n");

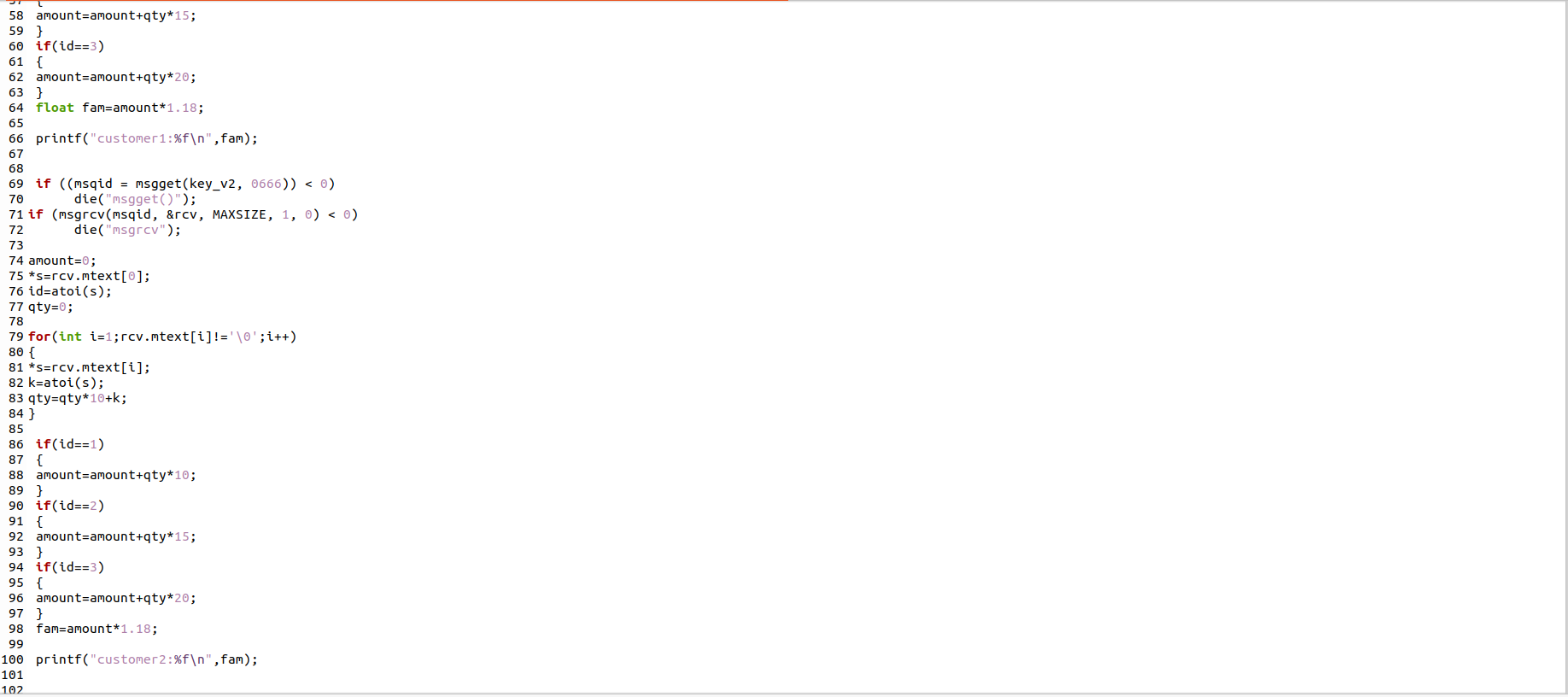
return 0;

}

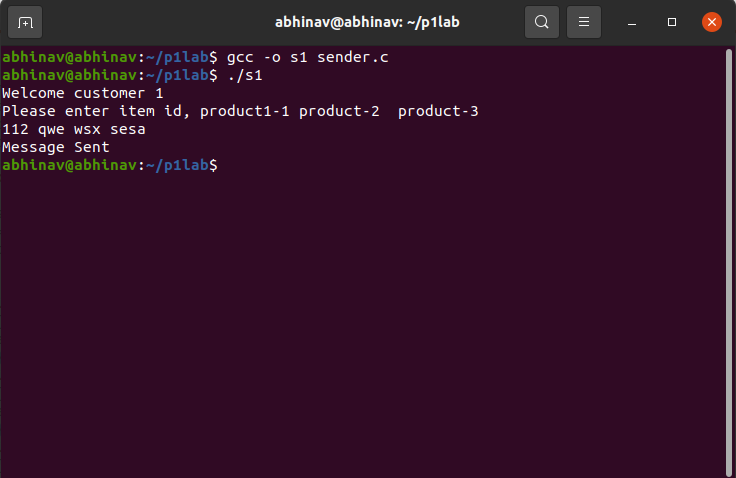
**Code Snippets:**

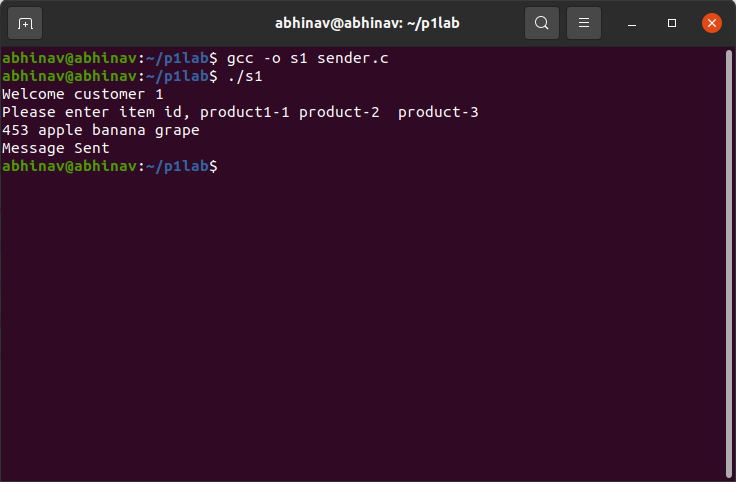
****

****

****

**Output :**

****

****