WEEK 6 LAB 6 :

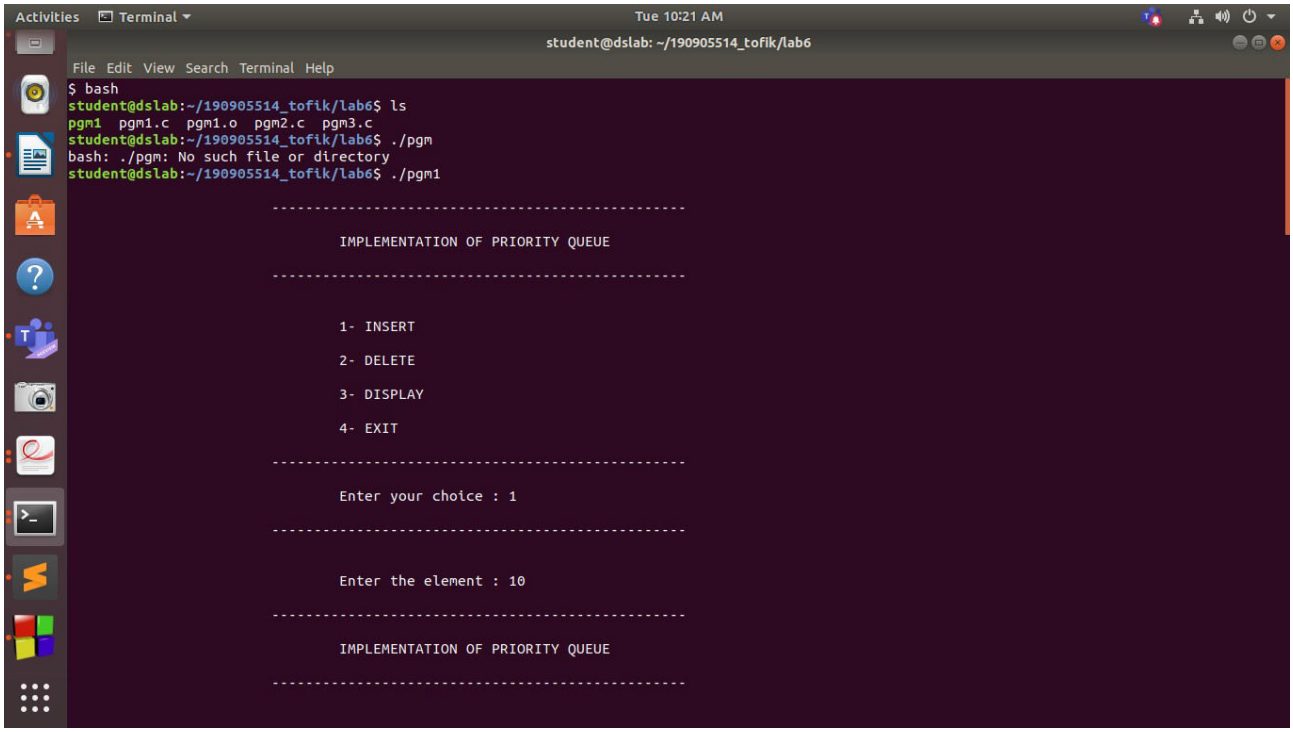
Q1. Implement an ascending priority queue.
Note: An ascending priority queue is a collection of items into which items can be inserted arbitrarily and from which only the smallest item can be removed. If apq is an ascending priority queue, the operation pqinsert(apq,x) inserts element x into apq and pqmindelete(apq) removes the minimum element from apq and returns its value.

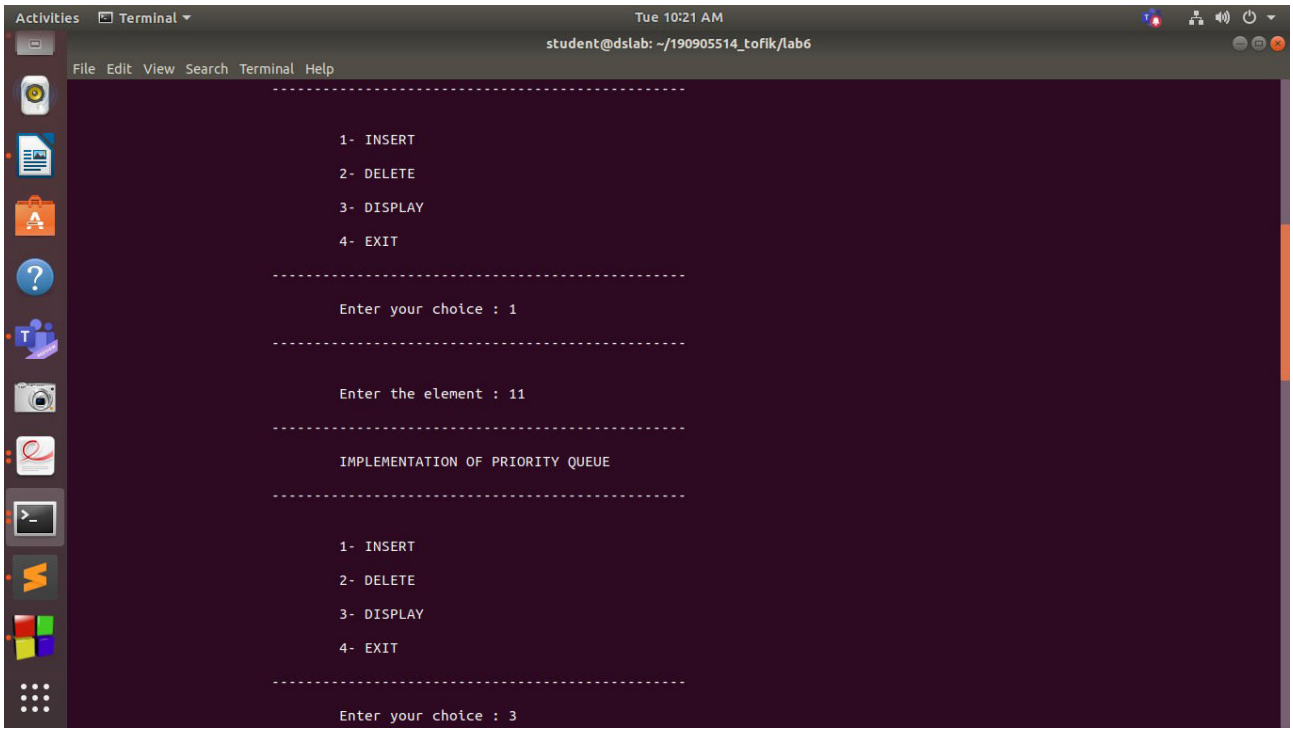
pgm1.c

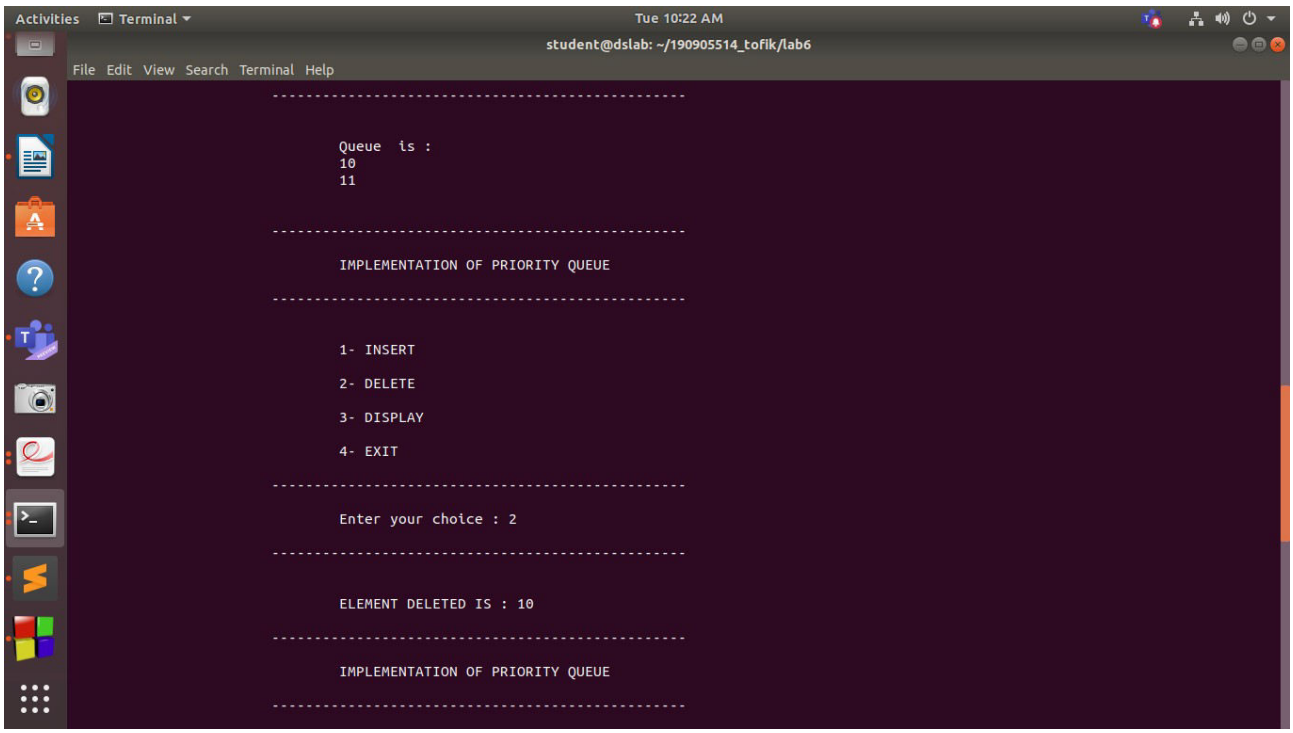
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAXIMUM 45
typedef struct{
        int QUEUE[MAXIMUM];
        int front;
        int rear;
}PQUEUE;

void insertPQ(PQUEUE *queue,int myName){
        int i,j;
        for(i=queue->front+1;i<=queue->rear;i++)
                if(queue->QUEUE[i]>myName)
                        break;
        if(i<=queue->rear){
                for(j=queue->rear;j>=i;j--)
                        queue->QUEUE[j+1]=queue->QUEUE[j];
        }
        queue->QUEUE[i]=myName;
        queue->rear++;
}

int deletePQ(PQUEUE *queue){
        if (queue->front==queue->rear){
                printf("\n\t\t\t\tQueue is Empty : \n\n");
                return -1;
        }
        return queue->QUEUE[++queue->front];
}

void displayPQ(PQUEUE *queue){
        if (queue->front==queue->rear){
                printf("\n\t\t\t\tQueue is Empty \n\n");
                return ;
        }
        int i;
        printf("\n\t\t\t\tQueue  is : ");
          printf("\n\n");
        for(i=queue->front+1;i<=queue->rear;i++)
```

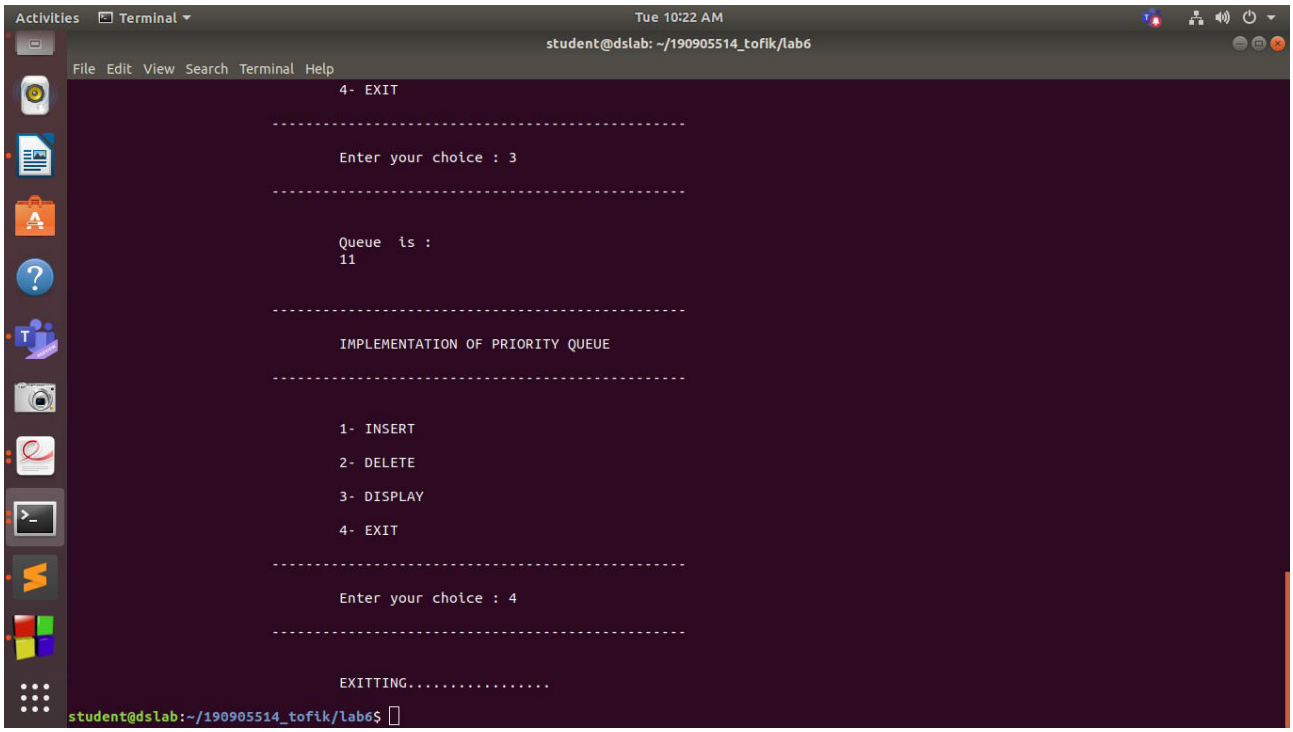
```
                printf("\n\t\t\t\t%d  ",queue->QUEUE[i]);
        printf("\n\n");
}

int main(){
        PQUEUE queue1,*queue;
        queue=&queue1;
        queue->front=queue->rear=-1;
   int myName,i,j;
   while(1)
   {
      int choice;
      printf("\n\t\t\t----------------------------------------------\n");
      printf("\n\t\t\t\tIMPLEMENTATION OF PRIORITY QUEUE\n");
      printf("\n\t\t\t----------------------------------------------\n\n");
      printf("\n\t\t\t\t1- INSERT\n");
      printf("\n\t\t\t\t2- DELETE\n");
      printf("\n\t\t\t\t3- DISPLAY\n");
      printf("\n\t\t\t\t4- EXIT\n");
      printf("\n\t\t\t----------------------------------------------\n");
      printf("\n\t\t\t\tEnter your choice : ");
      scanf("%d",&choice);
      printf("\n\t\t\t----------------------------------------------\n\n");
      switch(choice){
         case 1:
                printf("\n\t\t\t\tEnter the element : ");
                scanf("%d",&myName);
                        insertPQ(queue,myName);
                        break;
                case 2:
                        j = deletePQ(queue);
                        if(j!=-1)printf("\n\t\t\t\tELEMENT DELETED IS : %d\n",j);
                        break;

                case 3:
           displayPQ(queue);
           break;

                case 4 :
                        printf("\n\t\t\t\tEXITTING.....\n\n");
                        exit(0);

                default:
                        printf("\n\t\t\t\tInvalid choice  ! \n\n");
                        break;



      }
   }
   return 0;
}
```

OUTPUT :


Activities
Terminal
Tue 10:21 AM
student@dslab: ~/190905514_tofik/lab6
File Edit View Search Terminal Help
$ bash
student@dslab:~/190905514_tofik/lab6$ ls
pgm1 pgm1.c pgm1.o pgm2.c pgm3.c
student@dslab:~/190905514_tofik/lab6$ ./pgm
bash: ./pgm: No such file or directory
student@dslab:~/190905514_tofik/lab6$ ./pgm1
IMPLEMENTATION OF PRIORITY QUEUE
1- INSERT
2- DELETE
3- DISPLAY
4- EXIT
Enter your choice : 1
Enter the element : 10
IMPLEMENTATION OF PRIORITY QUEUE


Activities
Terminal
Tue 10:21 AM
student@dslab: ~/190905514_tofik/lab6
File Edit View Search Terminal Help
1- INSERT
2- DELETE
3- DISPLAY
4- EXIT
Enter your choice : 1
Enter the element : 11
IMPLEMENTATION OF PRIORITY QUEUE
1- INSERT
2- DELETE
3- DISPLAY
4- EXIT
Enter your choice : 3


Activities
Terminal
Tue 10:22 AM
student@dslab: ~/190905514_tofik/lab6
File Edit View Search Terminal Help
Queue is :
10
11
IMPLEMENTATION OF PRIORITY QUEUE
1- INSERT
2- DELETE
3- DISPLAY
4- EXIT
Enter your choice : 2
ELEMENT DELETED IS : 10
IMPLEMENTATION OF PRIORITY QUEUE


Activities
Terminal
Tue 10:22 AM
student@dslab: ~/190905514_tofik/lab6
File Edit View Search Terminal Help
4- EXIT
Enter your choice : 3
Queue is :
11
IMPLEMENTATION OF PRIORITY QUEUE
1- INSERT
2- DELETE
3- DISPLAY
4- EXIT
Enter your choice : 4
EXITTING.....
student@dslab:~/190905514_tofik/lab6$

Q.2 Implement a queue of strings using an output restricted dequeue (no deleteRight).
Note: An output-restricted deque is one where insertion can be made at both ends, but deletion can be made from one end only, where as An input-restricted deque is one where deletion can be made from both ends, but insertion can be made at one end only.

pgm2.c

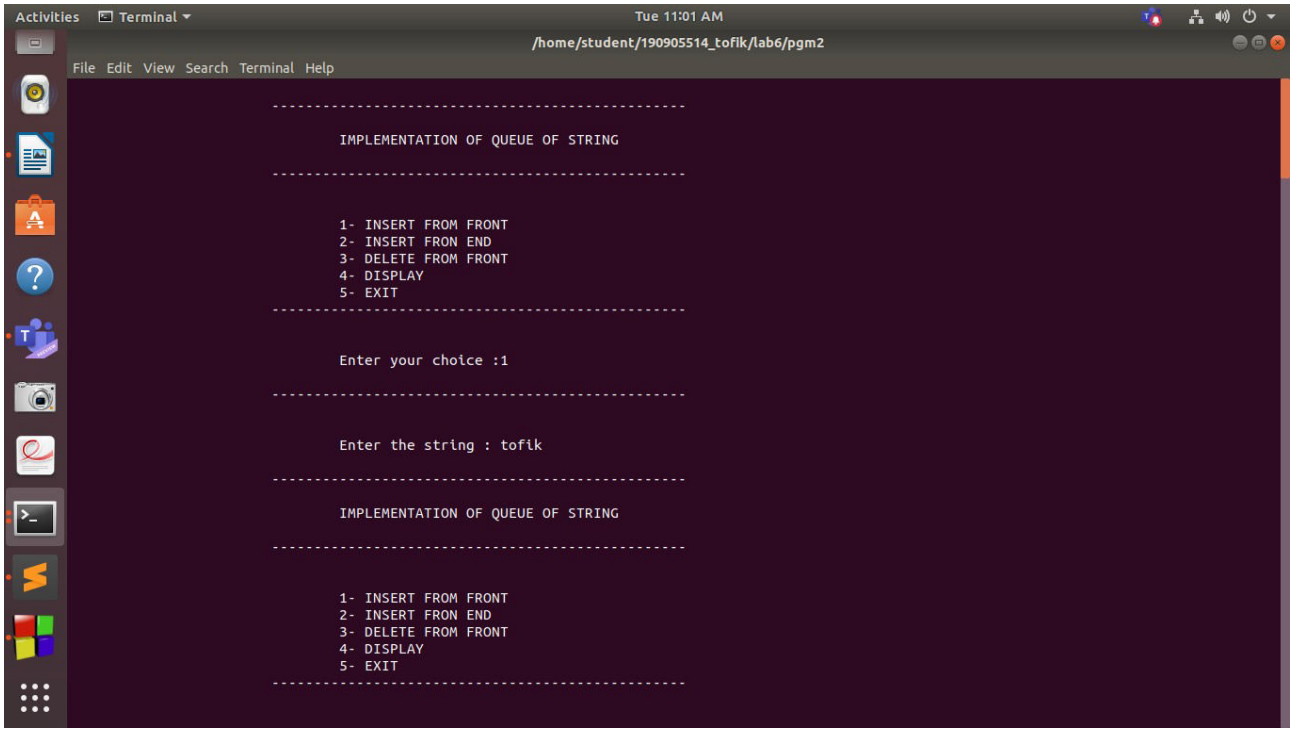
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAXIMUM 60
#define STRING 26
typedef struct
{
   char queue[MAXIMUM][STRING];
   int front;
   int rear;
} QUEUE;
void insertf(QUEUE *q,char ch[]);
void insertend(QUEUE *q,char ch[]);
void deletef(QUEUE *q);
void display(QUEUE *q);
int main()
{
   char myName[STRING],j;
   QUEUE *q,q1;
   q = &q1;
   q1.front = -1;
   q1.rear = -1;
   while(1)
   {
      int choice;
      printf("\n\t\t\t----------------------------------------------\n");
      printf("\n\t\t\t\tIMPLEMENTATION OF CIRCULAR QUEUE\n");
      printf("\n\t\t\t----------------------------------------------\n\n");
      printf("\n\t\t\t\t1- INSERT FROM FRONT");
      printf("\n\t\t\t\t2- INSERT FRON END");
      printf("\n\t\t\t\t3- DELETE FROM FRONT");
      printf("\n\t\t\t\t4- DISPLAY");
      printf("\n\t\t\t\t5- EXIT");
      printf("\n\t\t\t----------------------------------------------\n\n");
      printf("\n\t\t\t\tEnter your choice :");
      scanf("%d",&choice);
      printf("\n\t\t\t----------------------------------------------\n\n");
      switch(choice)
      {
      case 1:
         printf("\n\t\t\t\tEnter the string : ");

         scanf("%s",myName);
```

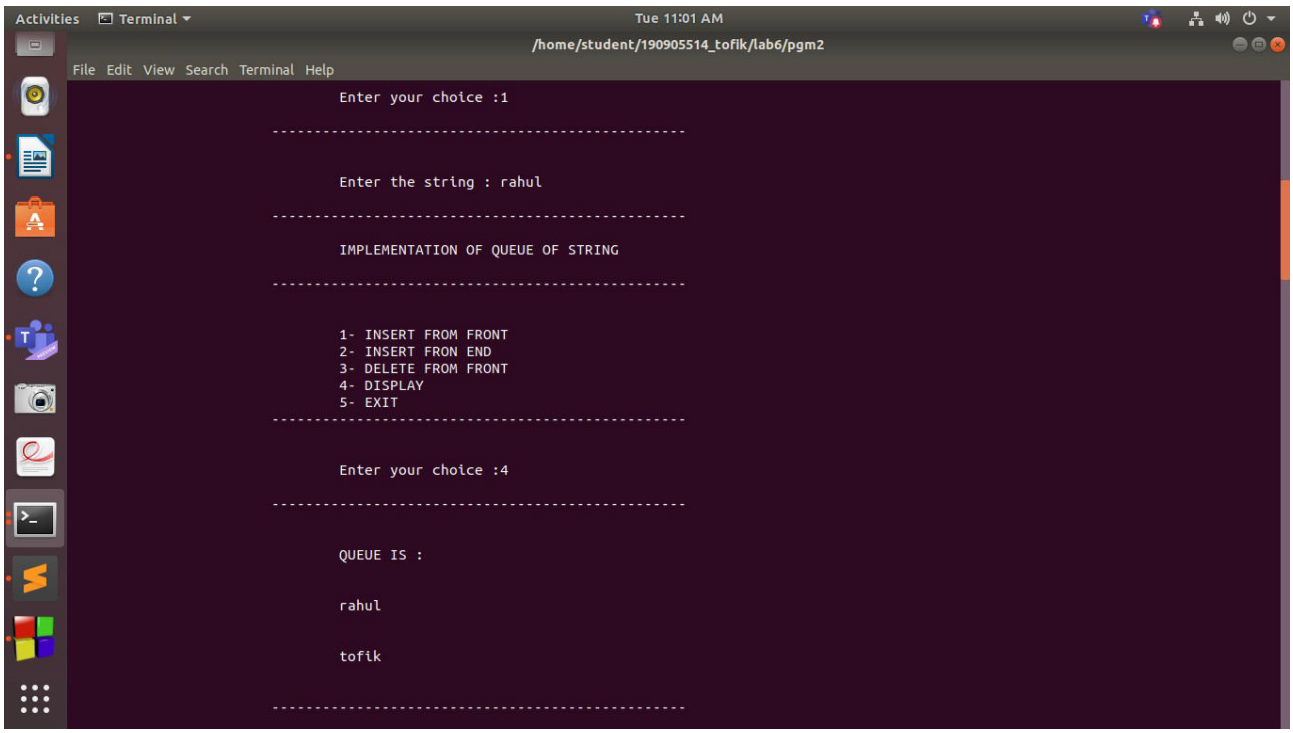
```
            insertf(&q1,myName);
            break;
        case 2:
            printf("\n\t\t\t\tEnter the string : ");

            scanf("%s",myName);

            insertend(&q1,myName);
            break;
        case 3:
            deletef(&q1);
            break;
        case 4:
            display(&q1);
            break;
        case 5:
            exit(0);
            break;
        default :
            printf("\n\t\t\t\tInvalid choice ! \n");
        }
    }
    return 0;
}

void insertf(QUEUE *q, char ch[])
{
    if(q->rear==MAXIMUM-1 && q->front==0)
    {
        printf("\n\t\t\t\tQueue is full : \n");
        return;
    }
    else if(q->front==-1)
    {
        q->front=q->rear=0;
        strcpy(q->queue[q->front],ch);
    }
    else if(q->rear!=MAXIMUM-1)
    {
        int i=q->rear+1;
        while(i>q->front)
        {
            strcpy(q->queue[i],q->queue[i-1]);
            i=i-1;
        }
        strcpy(q->queue[q->front],ch);

        q->rear++;
    }
    else
    {
        q->front--;
        strcpy(q->queue[q->front],ch);
    }
```

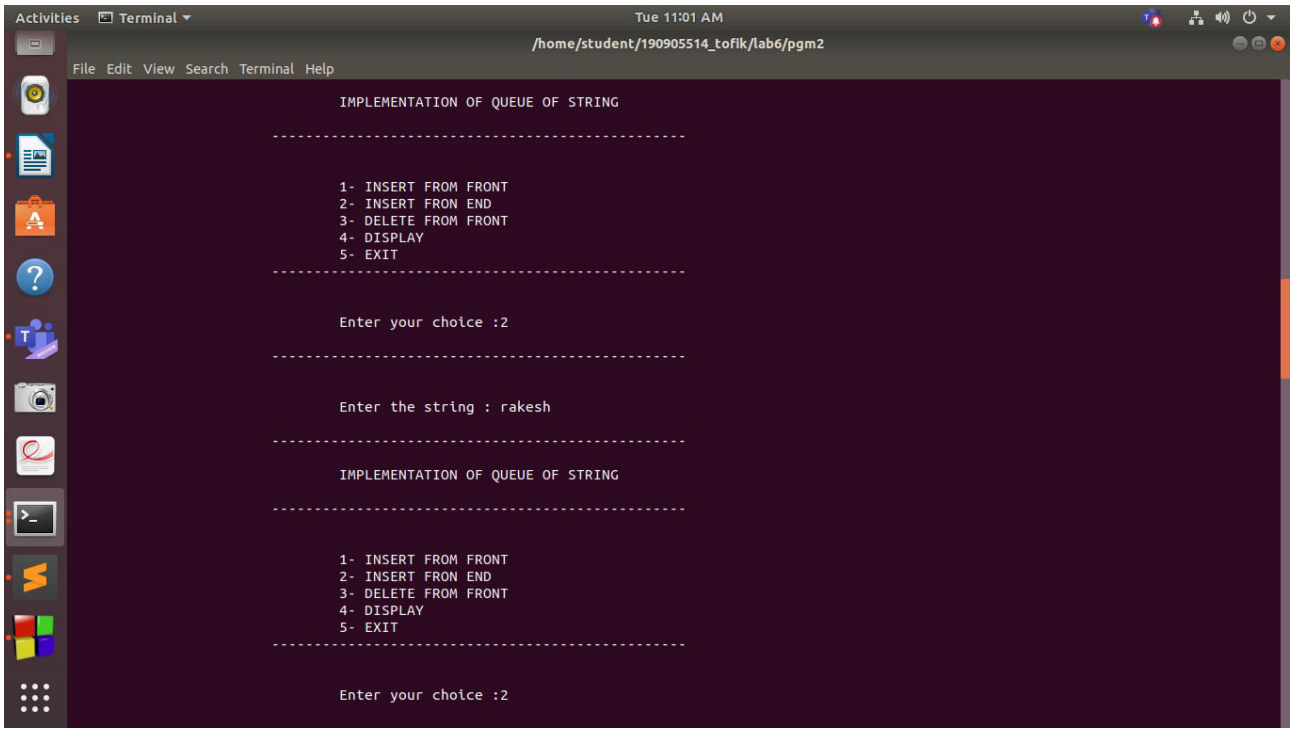
```
    //printf("%d and %d\n",q->front,q->rear);
    //printf("%s and %s\n",q->queue[q->front],q->queue[q->rear]);
}

void insertend(QUEUE *q, char ch[])
{
    if(q->rear==MAXIMUM-1 && q->front==0)
    {
        printf("\n\t\t\t\tQueue is full : \n");
        return;
    }
    else if(q->front==-1)
    {
        q->front=q->rear=0;
        strcpy(q->queue[q->rear],ch);
    }
    else if(q->front!=0)
    {
        int i=q->front-1;
        while(i<q->rear)
        {
            strcpy(q->queue[i],q->queue[i+1]);
            i=i+1;
        }
        strcpy(q->queue[q->rear],ch);
        q->front--;
    }
    else
    {
        q->rear++;
        strcpy(q->queue[q->rear],ch);
    }
    //printf("%d and %d\n",q->front,q->rear);
    //printf("%s and %s\n", q->queue[q->front], q->queue[q->rear]);
}

void deletef(QUEUE *q)
{
    if(q->front==-1)
    {

        printf("\n\t\t\t\tQueue is empty : \n");
        return ;
    }
    char ch[STRING] ;
    strcpy(ch, q->queue[q->front]);
    if(q->front==q->rear)
        q->front=q->rear=-1;
    else
        q->front++;
    printf("\n\t\t\t\t%s \n",ch);

}
```

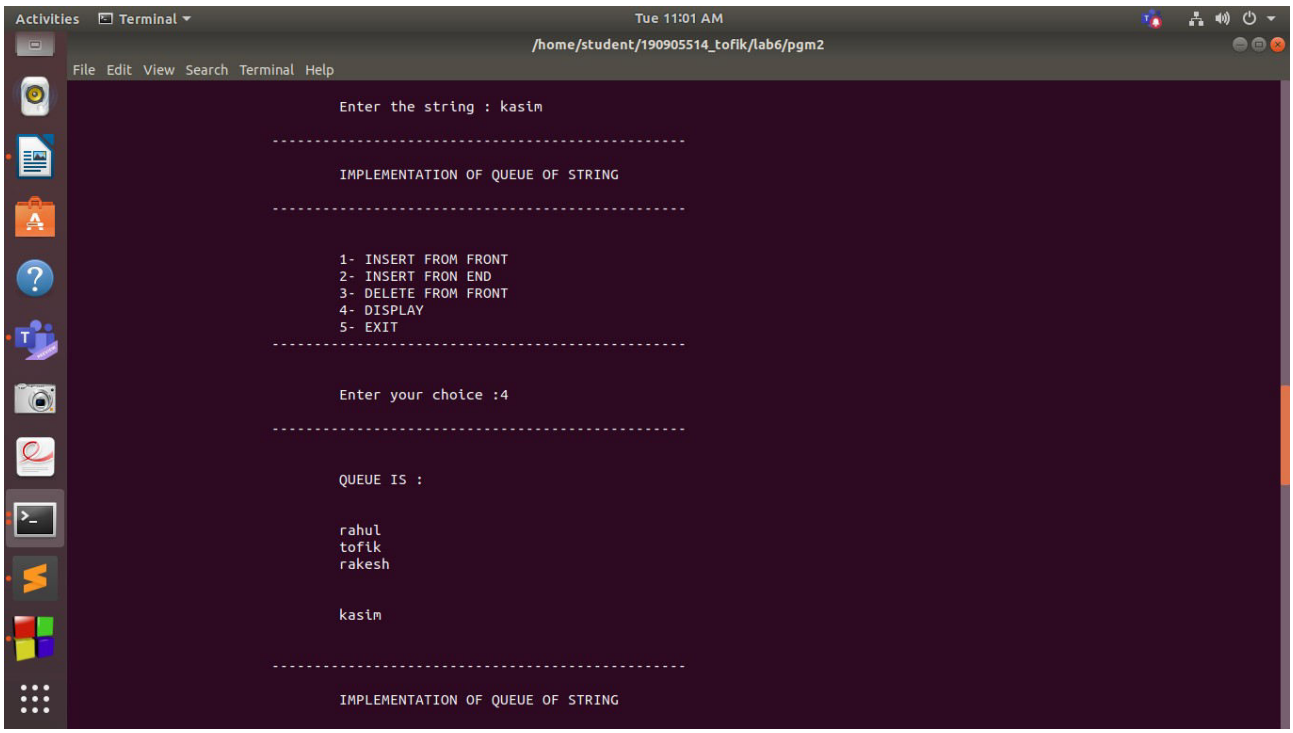
```
void display(QUEUE *q)
{
   if(q->front==-1)
   {
      printf("\n\t\t\t\tQueue is empty : \n");
      return ;
   }
   int i;
   printf("\n\t\t\t\tQUEUE IS : \n\n");
   for(i=q->front; i!=q->rear; i++)
   {
      printf("\n\t\t\t\t%s  ",q->queue[i]);
   }
   printf("\n\n");
   printf("\n\t\t\t\t%s  ",q->queue[i]);
   printf("\n\n");
}
```

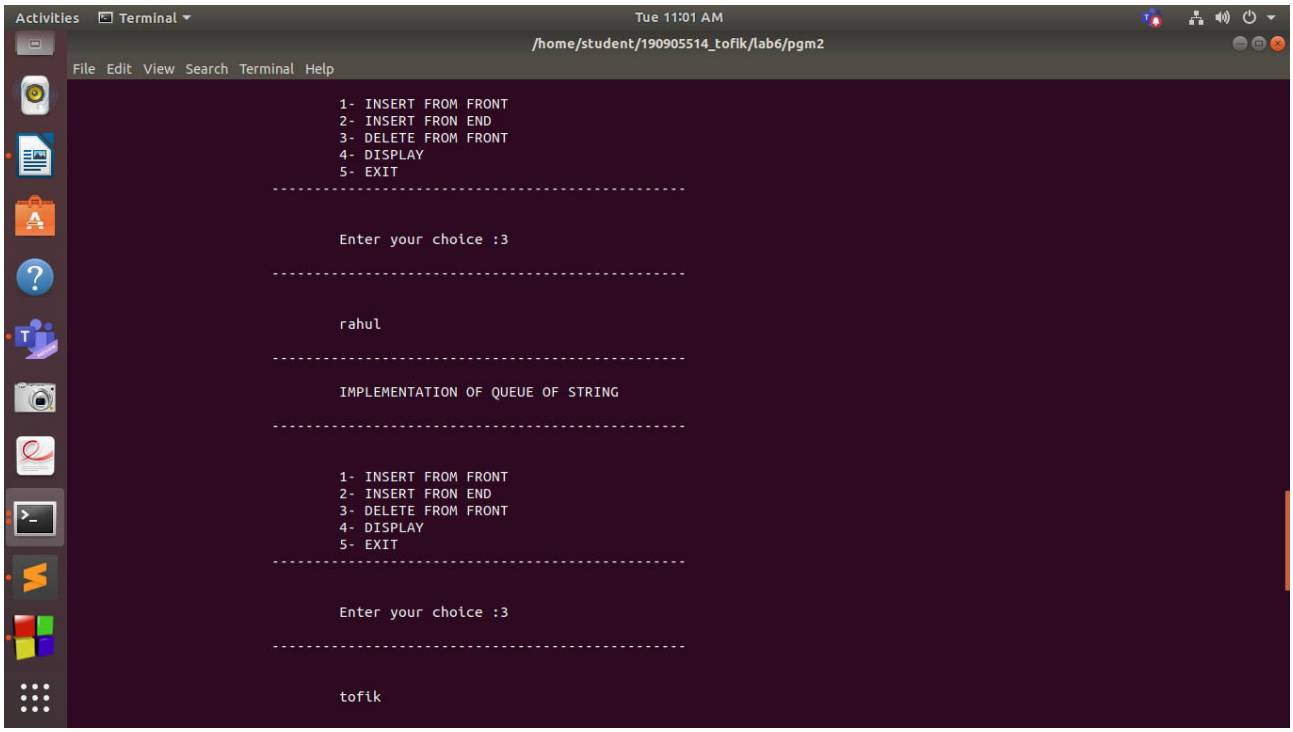
OUTPUT :

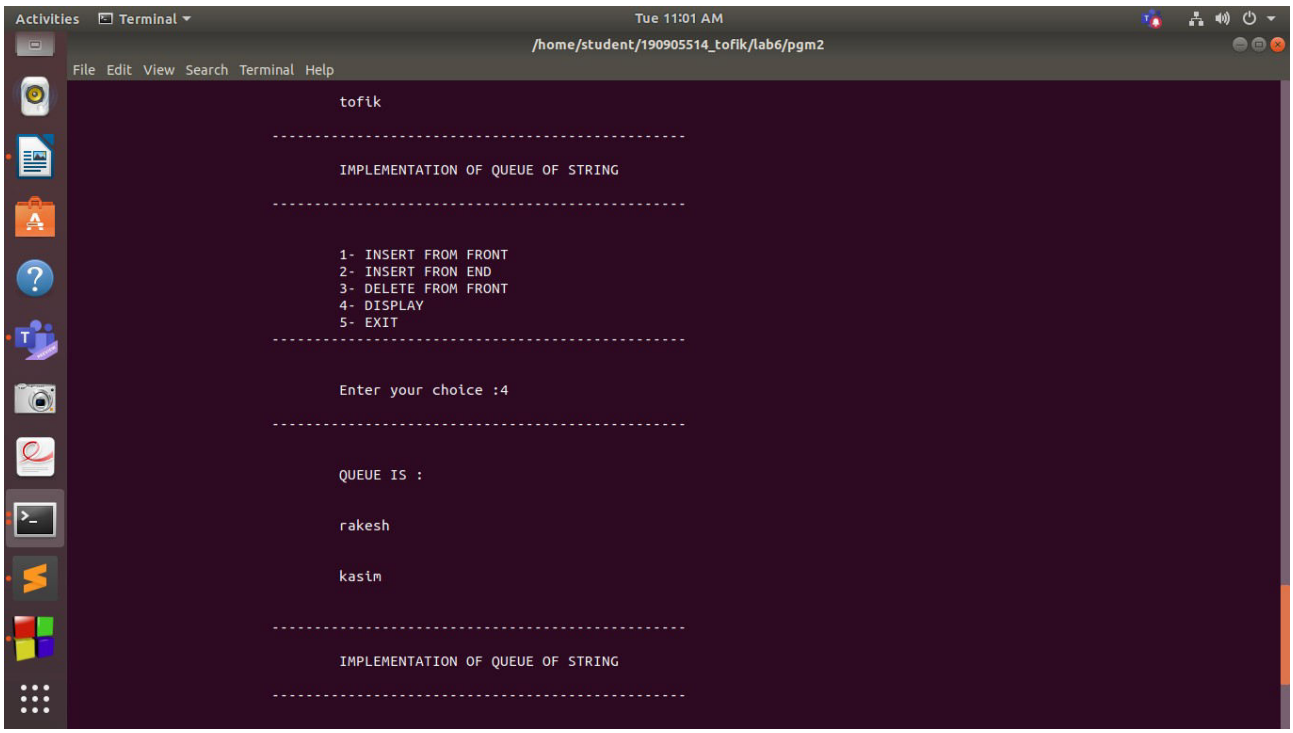


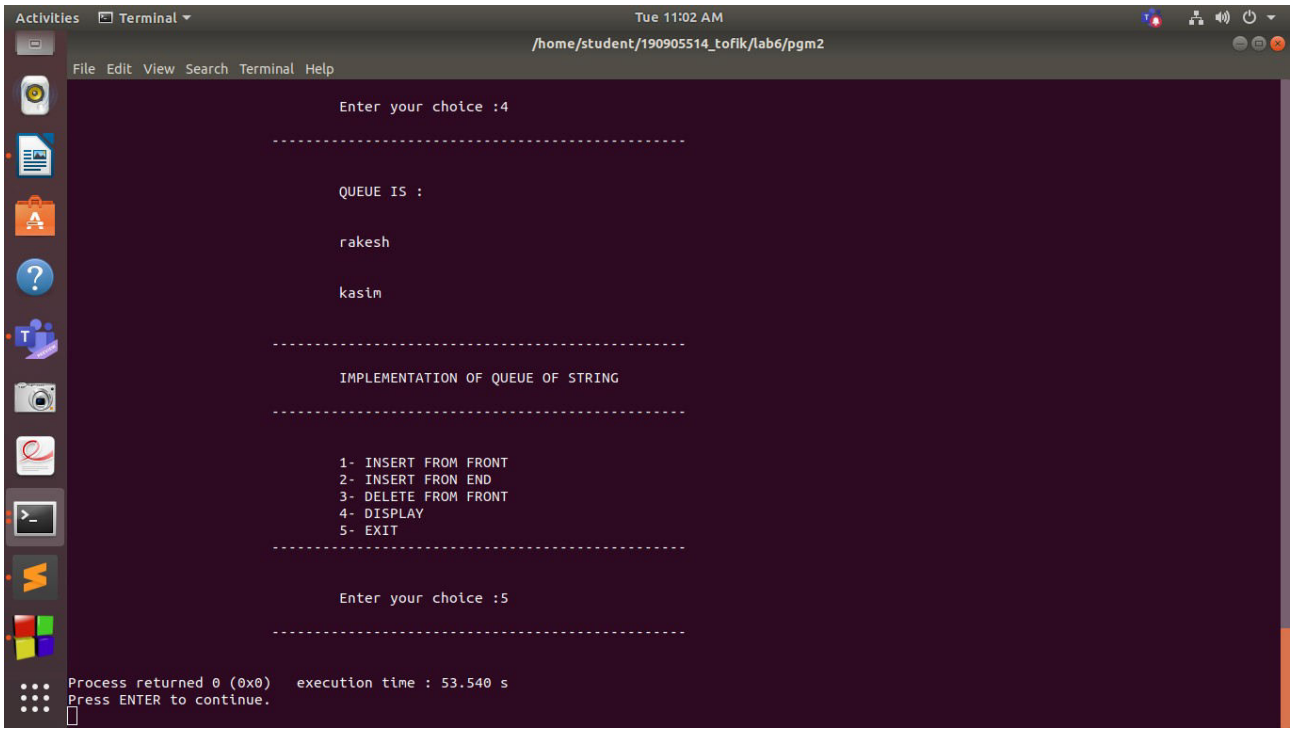

Activities
Terminal
Tue 11:01 AM
/home/student/190905514_tofik/lab6/pgm2
File Edit View Search Terminal Help
Enter your choice :1
Enter the string : rahul
IMPLEMENTATION OF QUEUE OF STRING
1- INSERT FROM FRONT
2- INSERT FRON END
3- DELETE FROM FRONT
4- DISPLAY
5- EXIT
Enter your choice :4
QUEUE IS :
rahul
tofik


Activities
Terminal
Tue 11:01 AM
/home/student/190905514_tofik/lab6/pgm2
File Edit View Search Terminal Help
IMPLEMENTATION OF QUEUE OF STRING
1- INSERT FROM FRONT
2- INSERT FRON END
3- DELETE FROM FRONT
4- DISPLAY
5- EXIT
Enter your choice :2
Enter the string : rakesh
IMPLEMENTATION OF QUEUE OF STRING
1- INSERT FROM FRONT
2- INSERT FRON END
3- DELETE FROM FRONT
4- DISPLAY
5- EXIT
Enter your choice :2


Activities
Terminal
Tue 11:01 AM
/home/student/190905514_tofik/lab6/pgm2
File Edit View Search Terminal Help
Enter the string : kasim
IMPLEMENTATION OF QUEUE OF STRING
1- INSERT FROM FRONT
2- INSERT FRON END
3- DELETE FROM FRONT
4- DISPLAY
5- EXIT
Enter your choice :4
QUEUE IS :
rahul
tofik
rakesh
kasim
IMPLEMENTATION OF QUEUE OF STRING


Activities
Terminal
Tue 11:01 AM
/home/student/190905514_tofik/lab6/pgm2
File Edit View Search Terminal Help
1- INSERT FROM FRONT
2- INSERT FRON END
3- DELETE FROM FRONT
4- DISPLAY
5- EXIT
Enter your choice :3
rahul
IMPLEMENTATION OF QUEUE OF STRING
1- INSERT FROM FRONT
2- INSERT FRON END
3- DELETE FROM FRONT
4- DISPLAY
5- EXIT
Enter your choice :3
tofik





Q.3 Write a program to check whether given string is a palindrome using a dequeue.

pgm3.c

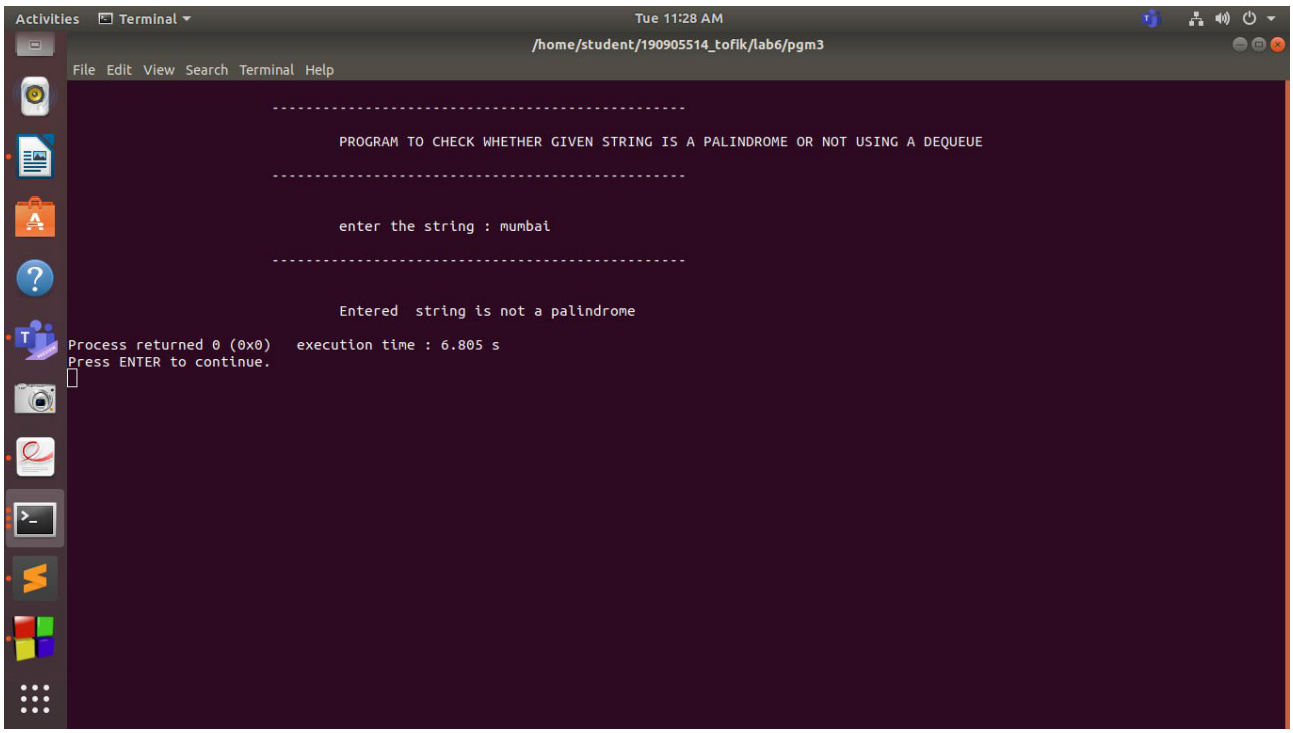
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
```

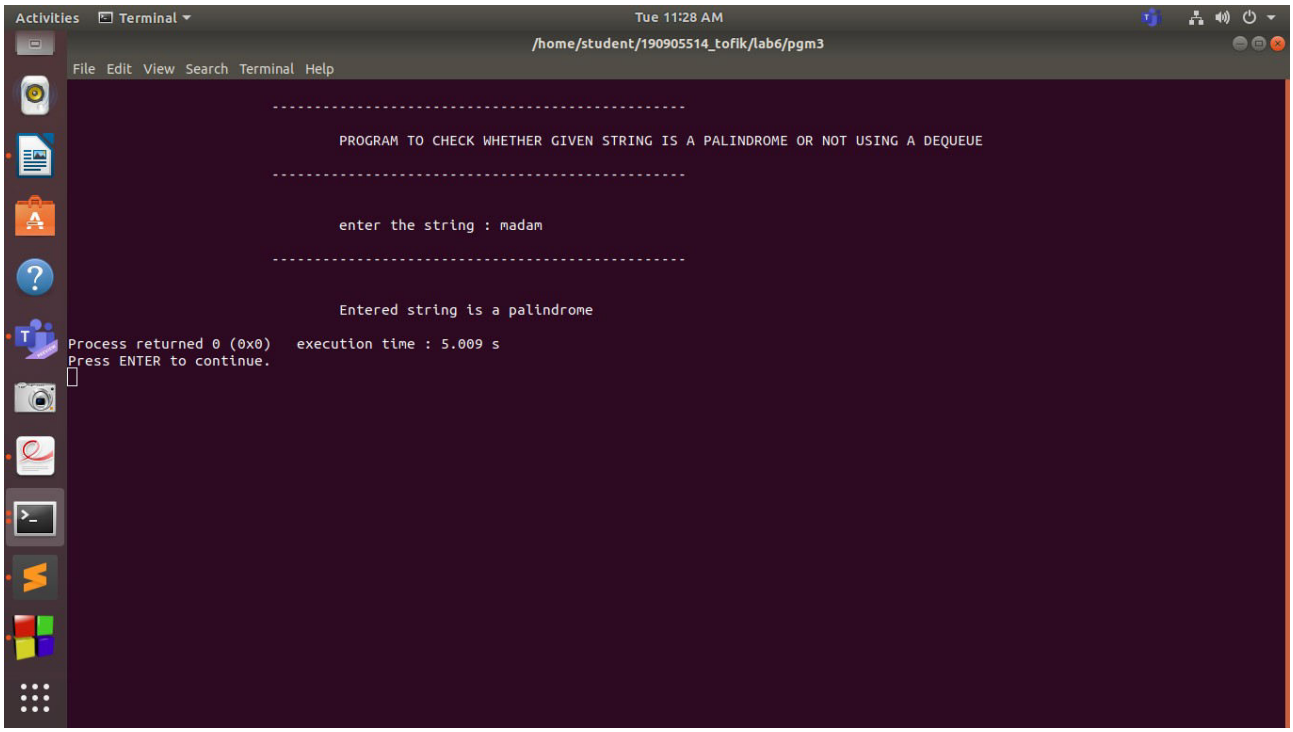
```
#define MAXIMUM 50

typedef struct
{
    int queue[MAXIMUM];
    int front;
    int rear;
} QUEUE;
void insertf(QUEUE *q,char c);
char deleteend(QUEUE *q);
char deletef(QUEUE *q);
void display(QUEUE *q);
int main()
{
    int i,j;
    QUEUE *q,q1;
    q = &q1;
    q1.front = -1;
    q1.rear = -1;
    char a,b;
    int s;
    char ch[20];
    printf("\n\t\t\t----------------------------------------------------\n");
    printf("\n\t\t\t\tPROGRAM TO CHECK WHETHER GIVEN STRING IS A PALINDROME OR NOT USING A
DEQUEUE\n");
    printf("\n\t\t\t----------------------------------------------------\n\n");
    printf("\n\t\t\t\tenter the string : ");
    scanf("%s",ch);
    printf("\n\t\t\t----------------------------------------------------\n\n");
    for(s=0; ch[s]!='\0'; s++)
    {
        insertf(q,ch[s]);
    }
    for(s=0; s<strlen(ch)/2; s++)
    {
        a=deleteend(q);
        b=deletef(q);
        if(a!=b)
        {
            printf("\n\t\t\t\tEntered  string is not a palindrome\n");
            exit(0);
        }
    }
    printf("\n\t\t\t\tEntered string is a palindrome\n");

    return 0;
}

void insertf(QUEUE *q, char ch)
{
    if(q->rear==MAXIMUM-1 && q->front==0)
    {
        printf("\n\t\t\t\tQueue is full FULL\n");
        return;
```

```
    }
    else if(q->front==-1)
    {
        q->front=q->rear=0;
        q->queue[q->front]=ch;
    }
    else if(q->rear!=MAXIMUM-1)
    {
        int s=q->rear+1;
        while(s>q->front)
        {
            q->queue[s]=q->queue[s-1];
            s=s-1;
        }
        q->queue[q->front]=ch;
        q->rear++;
    }
    else
    {
        q->front--;
        q->queue[q->front]=ch;
    }

}
char deleteend(QUEUE *q)
{
    if(q->front==-1)
    {
        printf("EMPTY\n");
        return -1;
    }
    int val = q->queue[q->rear];
    if(q->front==q->rear)
        q->front=q->rear=-1;
    else
        q->rear--;
    return val;
}
char deletef(QUEUE *q)
{
    if(q->front==-1)
    {
        printf("\n\t\t\t\tQueue is empty : \n\n");
        return -1;
    }
    int val = q->queue[q->front];
    if(q->front==q->rear)
        q->front=q->rear=-1;
    else
        q->front++;
    return val;
}
void display(QUEUE *q)
{
```

```
    if(q->front==-1)
    {
        printf("\n\t\t\t\tQueue is empty : \n");
        return ;
    }
    int i;
    printf("\n\t\t\t\tQUEUE IS : \n\n");
    for(i=q->front; i!=q->rear; i++)
    {
        printf("\n\t\t\t\t%d  ",q->queue[i]);
    }
    printf("\n\n");
    printf("\n\t\t\t\t%d  ",q->queue[i]);
    printf("\n\n");
}
```

OUTPUT :


Activities
Terminal
Tue 11:28 AM
/home/student/190905514_tofik/lab6/pgm3
File Edit View Search Terminal Help
PROGRAM TO CHECK WHETHER GIVEN STRING IS A PALINDROME OR NOT USING A DEQUEUE
enter the string : mumbai
Entered string is not a palindrome
Process returned 0 (0x0) execution time : 6.805 s
Press ENTER to continue.


Activities
Terminal
Tue 11:28 AM
/home/student/190905514_tofik/lab6/pgm3
File Edit View Search Terminal Help
PROGRAM TO CHECK WHETHER GIVEN STRING IS A PALINDROME OR NOT USING A DEQUEUE
enter the string : madam
Entered string is a palindrome
Process returned 0 (0x0) execution time : 5.009 s
Press ENTER to continue.