Name: Sohansingh Rajput
Roll no.:46
SY-IT

Program

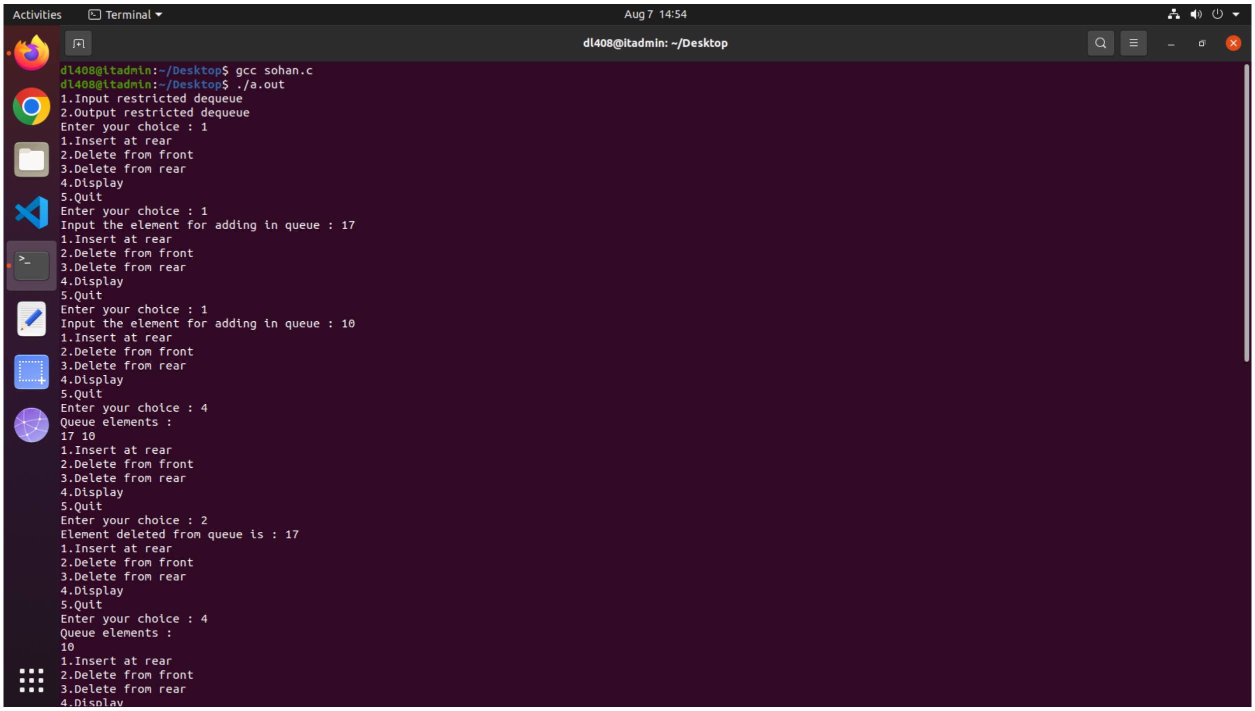
```
# include<stdio.h>
# define Size 3

int deque_arr[Size];
int front = -1;
int rear = -1;

/*Begin of insert_rear*/
void insert_rear()
{
        int added_item;
        if((front == 0 && rear == Size-1) || (front == rear+1))
        {   printf("Queue Overflow\n");
        return;}
        if (front == -1)  /* if queue is initially empty */
        {   front = 0;
        rear = 0;}
        else
        if(rear == Size-1)  /*rear is at last position of queue */
        rear = 0;
        else
        rear = rear+1;

        printf("Input the element for adding in queue : ");
        scanf("%d", &added_item);
        deque_arr[rear] = added_item ;
}
/*End of insert_rear*/

/*Begin of insert_front*/
void insert_front()
{   int added_item;
        if((front == 0 && rear == Size-1) || (front == rear+1))
        {   printf("Queue Overflow \n");
        return;  }
```

```
	if (front == -1)/*If queue is initially empty*/
	{   front = 0;
	rear = 0;          }
	else
	if(front== 0)
	front=Size-1;
	else
	front=front-1;
	printf("Input the element for adding in queue : ");
	scanf("%d", &added_item);
	deque_arr[front] = added_item ;  }
/*End of insert_front*/

/*Begin of delete_front*/
void delete_front()
{   if (front == -1)
	{   printf("Queue Underflow\n");
	return ;
	}
	printf("Element deleted from queue is : %d\n",deque_arr[front]);
	if(front == rear) /*Queue has only one element */
	{   front = -1;
	rear=-1;
	}
	else
	if(front == Size-1)
	front = 0;
	else
	front = front+1;
}
/*End of delete_front*/

/*Begin of delete_rear*/
void delete_rear()
{
	if (front == -1)
	{
	printf("Queue Underflow\n");
	return ;
	}
	printf("Element deleted from queue is : %d\n",deque_arr[rear]);
	if(front == rear) /*queue has only one element*/
	{
	front = -1;
```

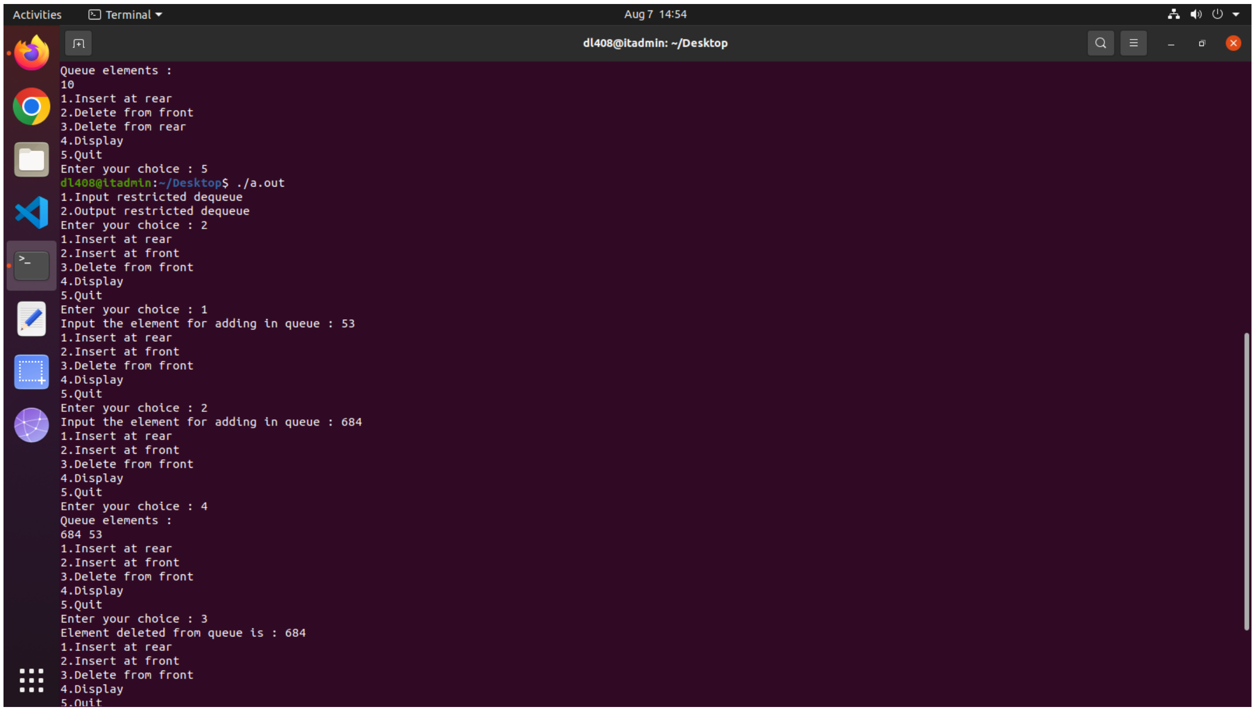
```
        rear=-1;
        }
        else
        if(rear == 0)
        rear=Size-1;
        else
        rear=rear-1;    }
/*End of delete_rear*/

/*Begin of input_que*/
void display_queue()
{
        int front_pos = front,rear_pos = rear;

        if(front == -1)
        {   printf("Queue is empty\n");
        return;
        }
        printf("Queue elements :\n");
        if( front_pos <= rear_pos )
        {
        while(front_pos <= rear_pos)
        {
        printf("%d ",deque_arr[front_pos]);
        front_pos++;
        }
        }
        else
        {
        while(front_pos <= Size-1)
        {   printf("%d ",deque_arr[front_pos]);
        front_pos++;
        }
        front_pos = 0;
        while(front_pos <= rear_pos)
        {
        printf("%d ",deque_arr[front_pos]);
        front_pos++;
        }
        }
        printf("\n");
}
/*End of display_queue*/
```

```
/*Begin of input_que*/
void input_que()
{   int choice;
        do
        {   printf("1.Insert at rear\n");
        printf("2.Delete from front\n");
        printf("3.Delete from rear\n");
        printf("4.Display\n");
        printf("5.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);

        switch(choice)
        {   case 1:
        insert_rear();
        break;
        case 2:
        delete_front();
        break;
        case 3:
        delete_rear();
        break;
        case 4:
        display_queue();
        break;
        case 5:
        break;
        default:
        printf("Wrong choice\n");
        }
        }while(choice!=5);
}
/*End of input_que*/

/*Begin of output_que*/
void output_que()
{   int choice;
        do
        {   printf("1.Insert at rear\n");
        printf("2.Insert at front\n");
        printf("3.Delete from front\n");
        printf("4.Display\n");
        printf("5.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
        insert_rear();
        break;
        case 2:
        insert_front();
        break;
        case 3:
        delete_front();
        break;
        case 4:
        display_queue();
        break;
        case 5:
        break;
        default:
        printf("Wrong choice\n");
        }
        }while(choice!=5);
}
/*End of output_que*/

/*Begin of main*/
int main()
{   int choice;
        printf("1.Input restricted dequeue\n");
        printf("2.Output restricted dequeue\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1 :
        input_que();
        break;
        case 2:
        output_que();
        break;
        default:
        printf("Wrong choice\n");
        }
}
```

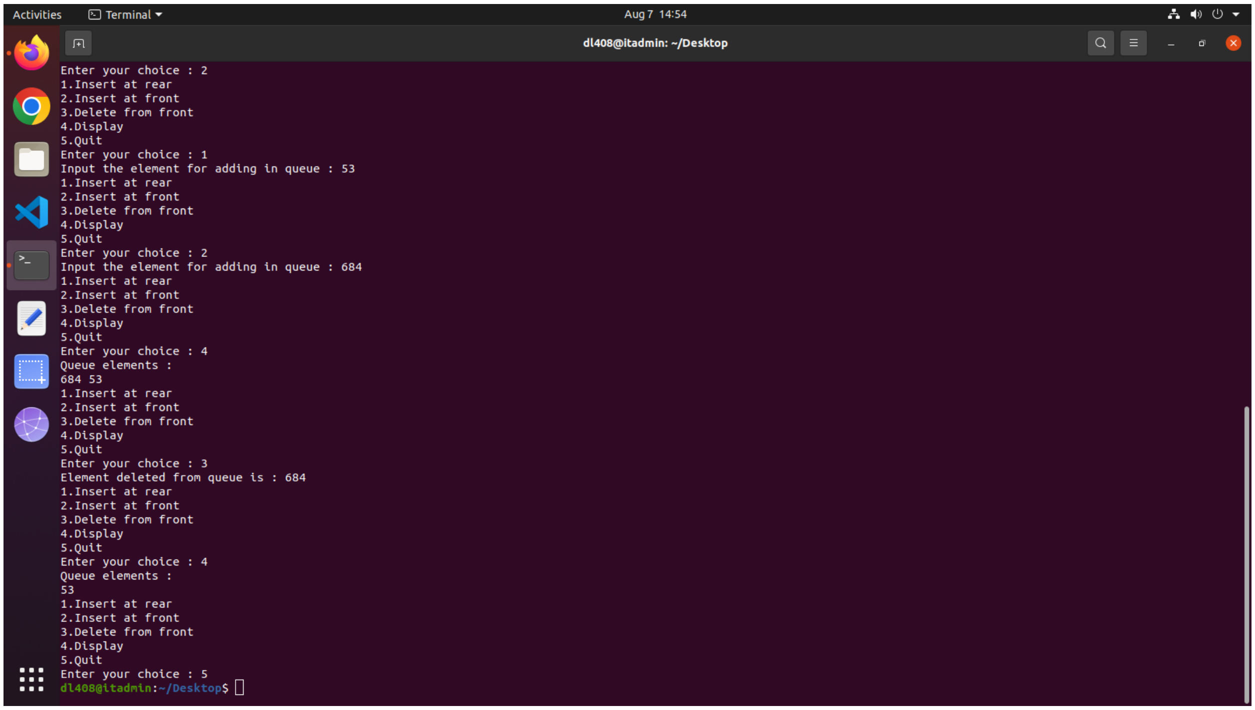
Output:



```
dl408@itadmin:~/Desktop$ gcc sohan.c
dl408@itadmin:~/Desktop$ ./a.out
1.Input restricted dequeue
2.Output restricted dequeue
Enter your choice : 1
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 1
Input the element for adding in queue : 17
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 1
Input the element for adding in queue : 10
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 4
Queue elements :
17 10
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 2
Element deleted from queue is : 17
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 4
Queue elements :
10
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
```



```
Queue elements :
10
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 5
dl408@itadmin:~/Desktop$ ./a.out
1.Input restricted dequeue
2.Output restricted dequeue
Enter your choice : 2
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 1
Input the element for adding in queue : 53
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 2
Input the element for adding in queue : 684
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 4
Queue elements :
684 53
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 3
Element deleted from queue is : 684
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
```



```
Enter your choice : 2
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 1
Input the element for adding in queue : 53
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 2
Input the element for adding in queue : 684
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 4
Queue elements :
684 53
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 3
Element deleted from queue is : 684
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 4
Queue elements :
53
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 5
dl408@itadmin:~/Desktop$ 
```