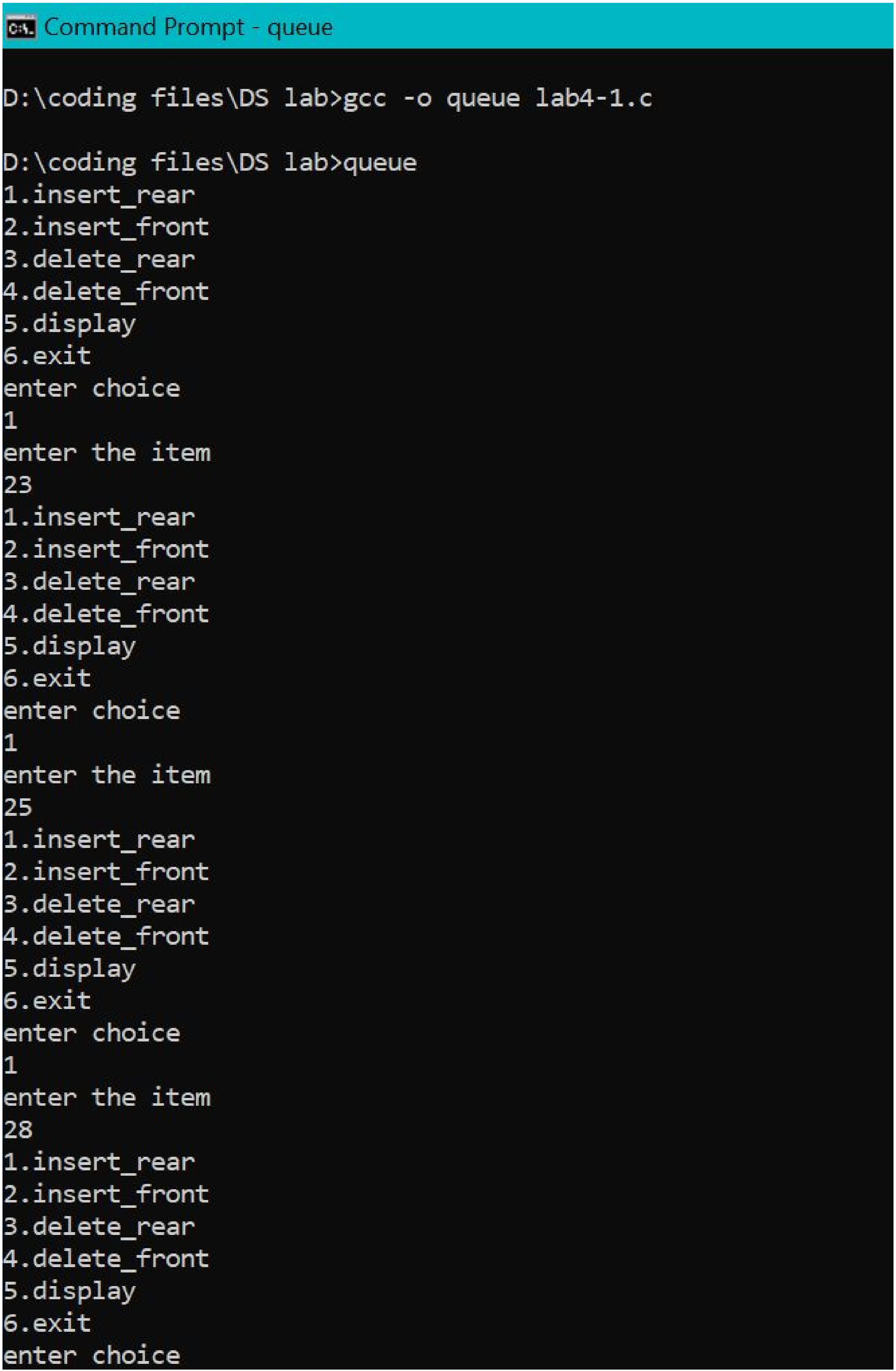
LAB-4

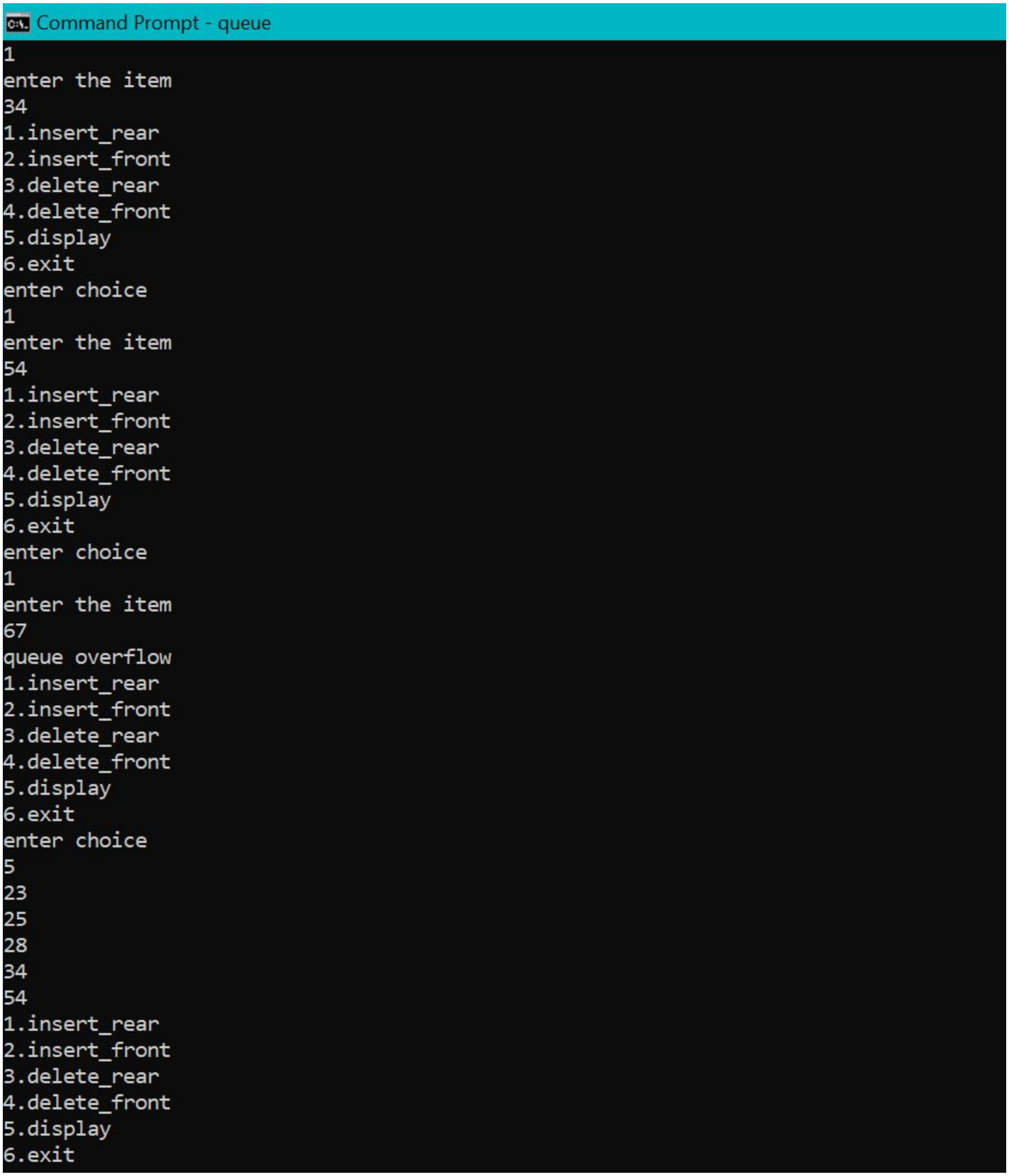
1.Double Ended Queue

```
#include<stdio.h>
#include<stdlib.h>
#define qsize 5
int f=0,r=-1,ch;
int item,q[10];

int isfull()
  {
   return(r==qsize-1)?1:0;
  }
int isempty()
  {
   return(f>r)?1:0;
  }
void insert_rear()
  {
   if(isfull())
            {
             printf("queue overflow\n");
             return;
            }
   r=r+1;
   q[r]=item;
  }
void delete_front()
  {
   if(isempty())
            {
             printf("queue empty\n");
             return;
            }
   printf("item deleted is %d\n",q[(f)++]);
   if(f>r)
            {
             f=0;
             r=-1;
            }
  }
void insert_front()
  {
   if(f!=0)
            {
             f=f-1;
             q[f]=item;
             return;
            }
           else if((f==0)&&(r==-1))
            {
             q[++(r)]=item;
             return;
            }
           else
             printf("insertion not possible\n");
  }
void delete_rear()
  {
   if(isempty())
            {
             printf("queue is empty\n");
             return;
            }
   printf("item deleted is %d\n",q[(r)--]);
   if(f>r)
            {
             f=0;
             r=-1;
            }
  }
void display()
  {
   int i;
   if(isempty())
            {
             printf("queue empty\n");
             return;
            }
   for(i=f;i<=r;i++)
           printf("%d\n",q[i]);
  }
```

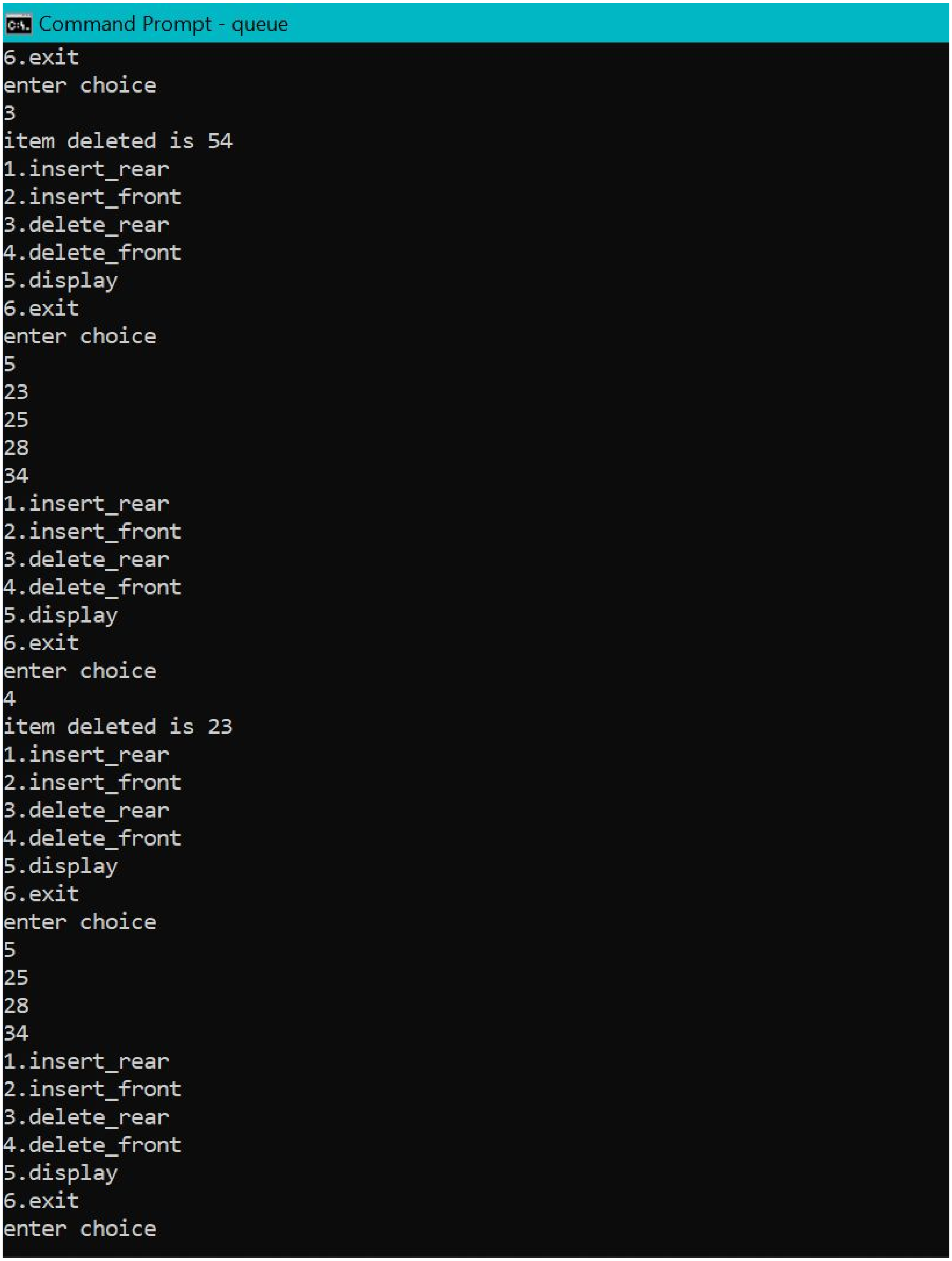
```
void main()
 {
  for(;;)
   {
          printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
          printf("enter choice\n");
          scanf("%d",&ch);
          switch(ch)
            {
             case 1:printf("enter the item\n");
                             scanf("%d",&item);
                             insert_rear();
                             break;
             case 2:printf("enter the item\n");
                             scanf("%d",&item);
                             insert_front();
                             break;
             case 3:delete_rear();
                             break;
             case 4:delete_front();
                             break;
             case 5:display();
                             break;
             default:exit(0);
            }
          }
  }
```



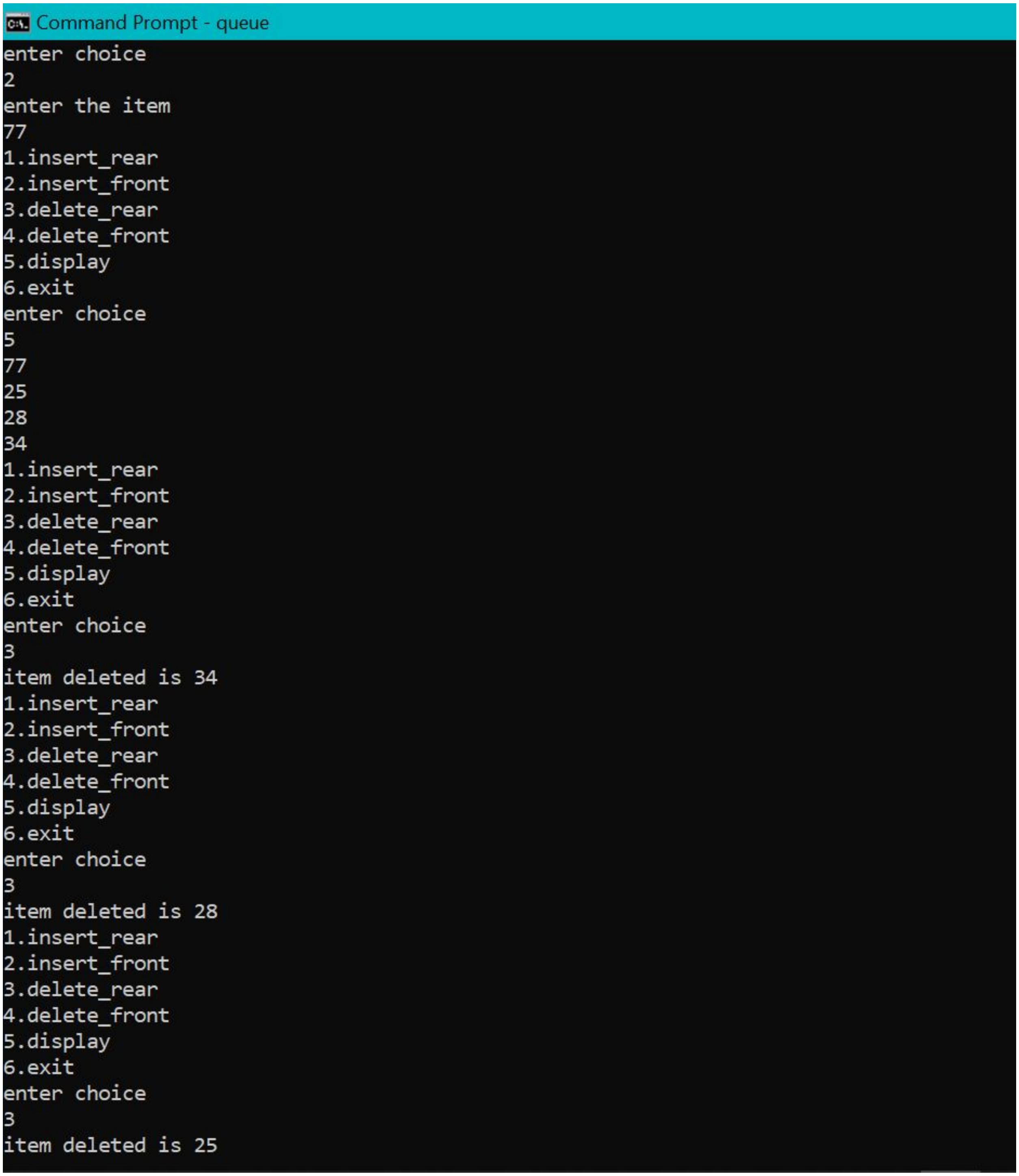
```
D:\coding files\DS lab>gcc -o queue lab4-1.c

D:\coding files\DS lab>queue
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
23
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
25
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
28
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```



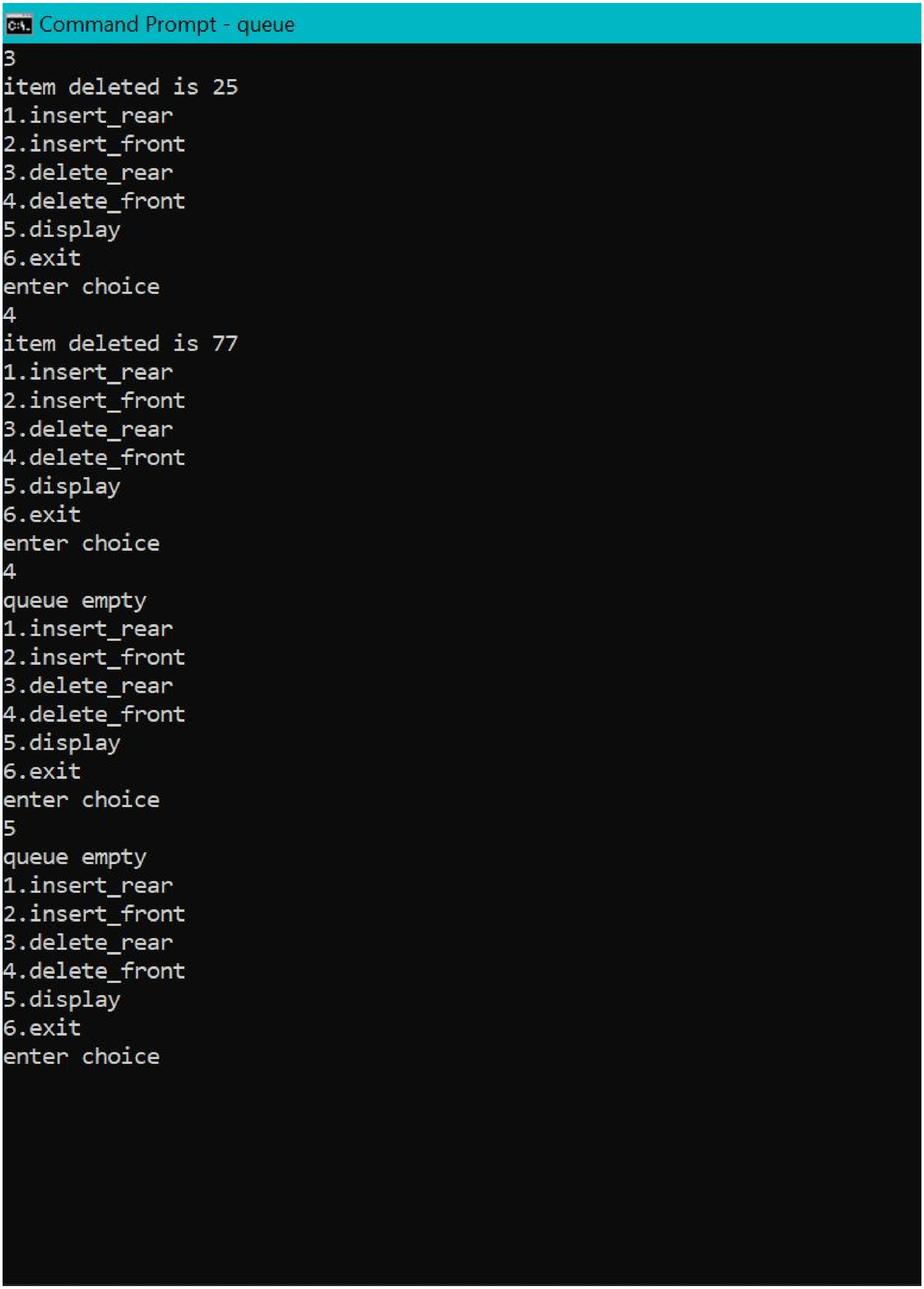
```
1
enter the item
34
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
54
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
67
queue overflow
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
23
25
28
34
54
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
```



```
6.exit
enter choice
3
item deleted is 54
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
23
25
28
34
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 23
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
25
28
34
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```



```
enter choice
2
enter the item
77
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
77
25
28
34
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 34
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 28
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 25
```



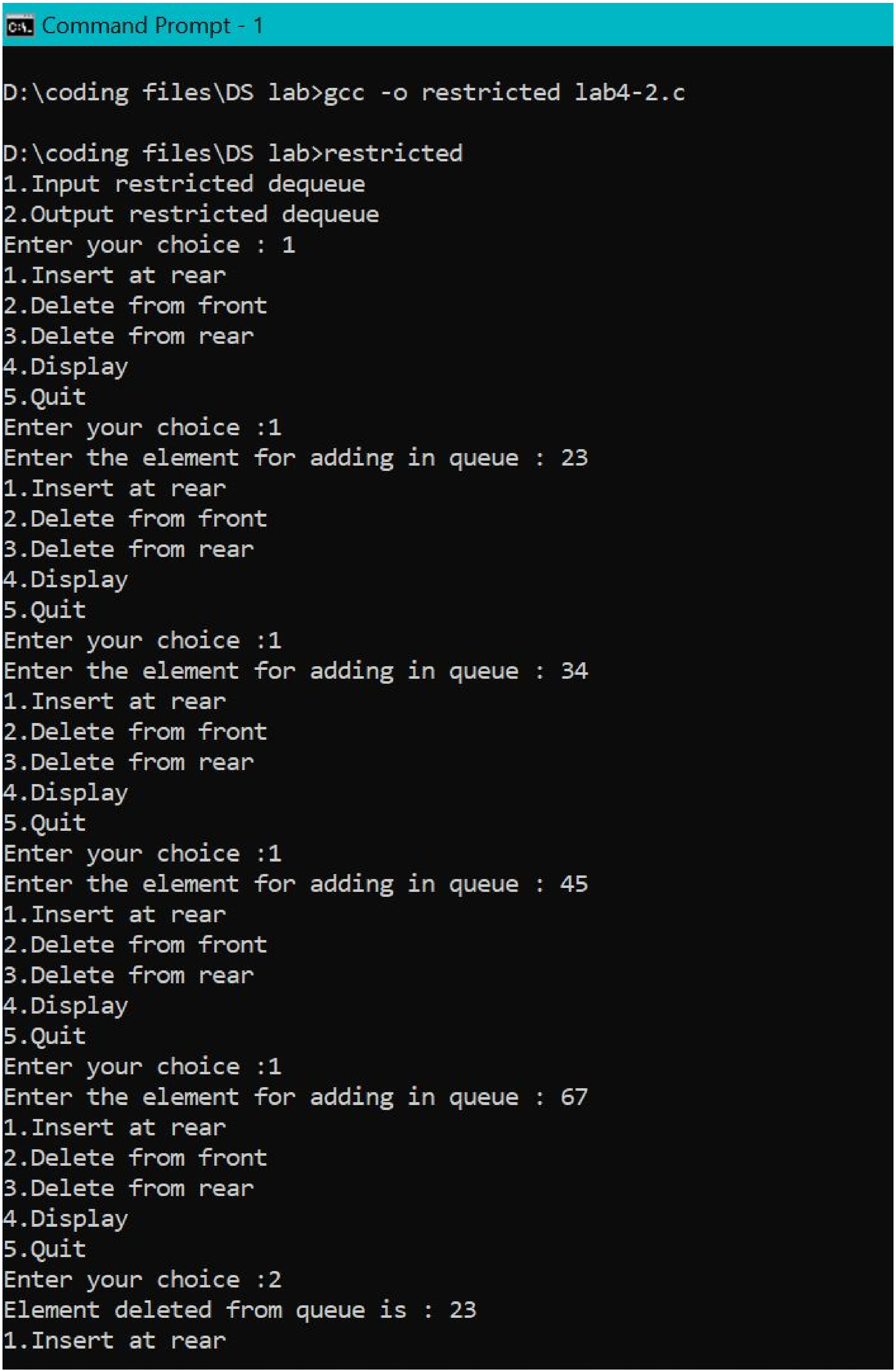
```
3
item deleted is 25
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 77
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
queue empty
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
queue empty
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```

2.Input and Output restricted queue

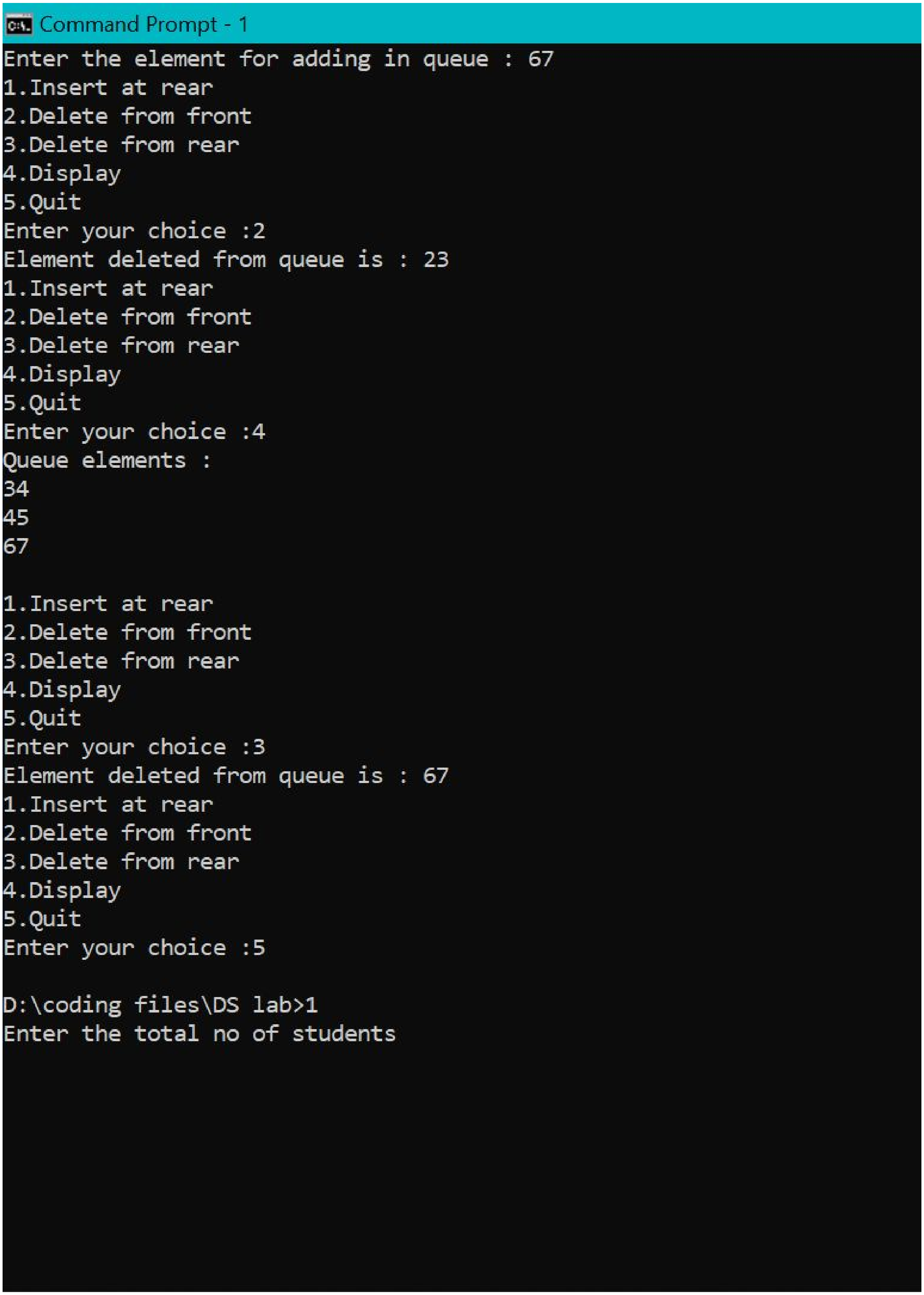
```
# include<stdio.h>
# define Size 5
int deque_arr[Size];
int front = -1,rear = -1;


void insert_rear()
{
   int added_item;
   if((front == 0 && rear == Size-1) || (front == rear+1))
   {   printf("***Queue Overflow***\n");
       return;
       }
   if (front == -1)
   {
       front = 0;
       rear = 0;
       }
   else
   if(rear == Size-1)
       rear = 0;
   else
       rear = rear+1;

   printf("Enter the element for adding in queue : ");
   scanf("%d",&added_item);
   deque_arr[rear] = added_item ;
}


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

```
        }
    else
    if(front== 0)
        front=Size-1;
    else
        front=front-1;
    printf("Enter the element for adding in queue : ");
    scanf("%d", &added_item);
    deque_arr[front] = added_item ;
    }



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)
    {   front = -1;
        rear=-1;
    }
    else
        if(front == Size-1)
            front = 0;
        else
            front = front+1;
}



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)
    {
        front = -1;
        rear=-1;
    }
    else
        if(rear == 0)
            rear=Size-1;
        else
            rear=rear-1;
            }



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 \n",deque_arr[front_pos]);
            front_pos++;
        }
    }
    else
    {
        while(front_pos <= Size-1)
        {   printf("%d \n",deque_arr[front_pos]);
            front_pos++;
        }
        front_pos = 0;
        while(front_pos <= rear_pos)
        {
            printf("%d \n",deque_arr[front_pos]);
            front_pos++;
        }
    }
    printf("\n");
}


/*Input Queue*/
```

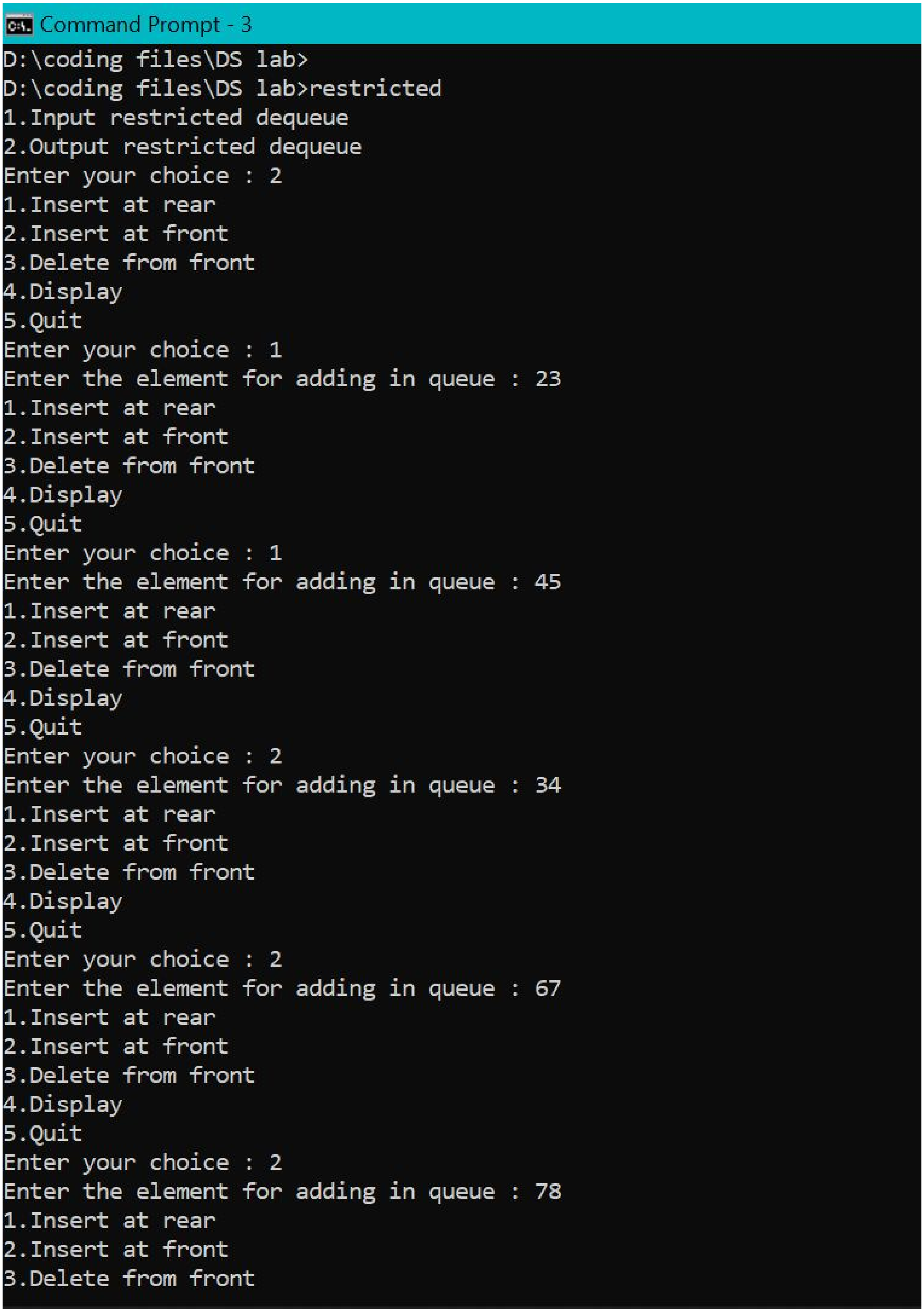
```
void input_que()
{   int choice;
    do
    {   printf("1.Insert at rear\n2.Delete from front\n3.Delete from rear\n4.Display\n5.Quit\n");
        printf("Enter your choice :");
        scanf("%d",&choice);
        switch(choice)
        {   case 1:
            insert_rear();
            break;
         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);
}


/*Output Queue*/
void output_que()
{   int choice;
    do
    {   printf("1.Insert at rear\n2.Insert at front\n3.Delete from front\n4.Display\n5.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);
}



main()
{   int choice;
    printf("1.Input restricted dequeue\n2.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");
    }
}
```



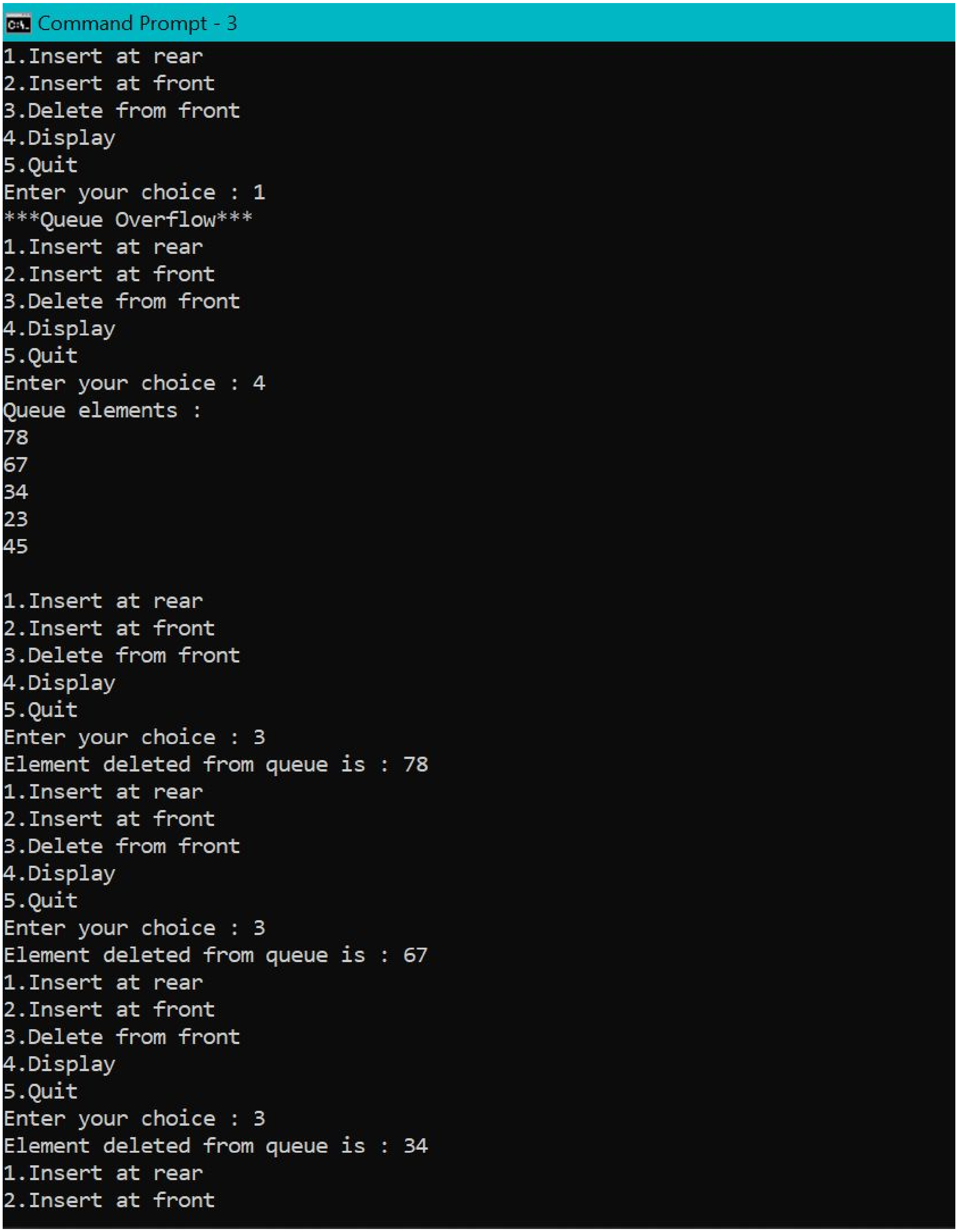
```
D:\coding files\DS lab>gcc -o restricted lab4-2.c

D:\coding files\DS lab>restricted
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
Enter the element for adding in queue : 23
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice :1
Enter the element for adding in queue : 34
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice :1
Enter the element for adding in queue : 45
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice :1
Enter the element for adding in queue : 67
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 : 23
1.Insert at rear
```



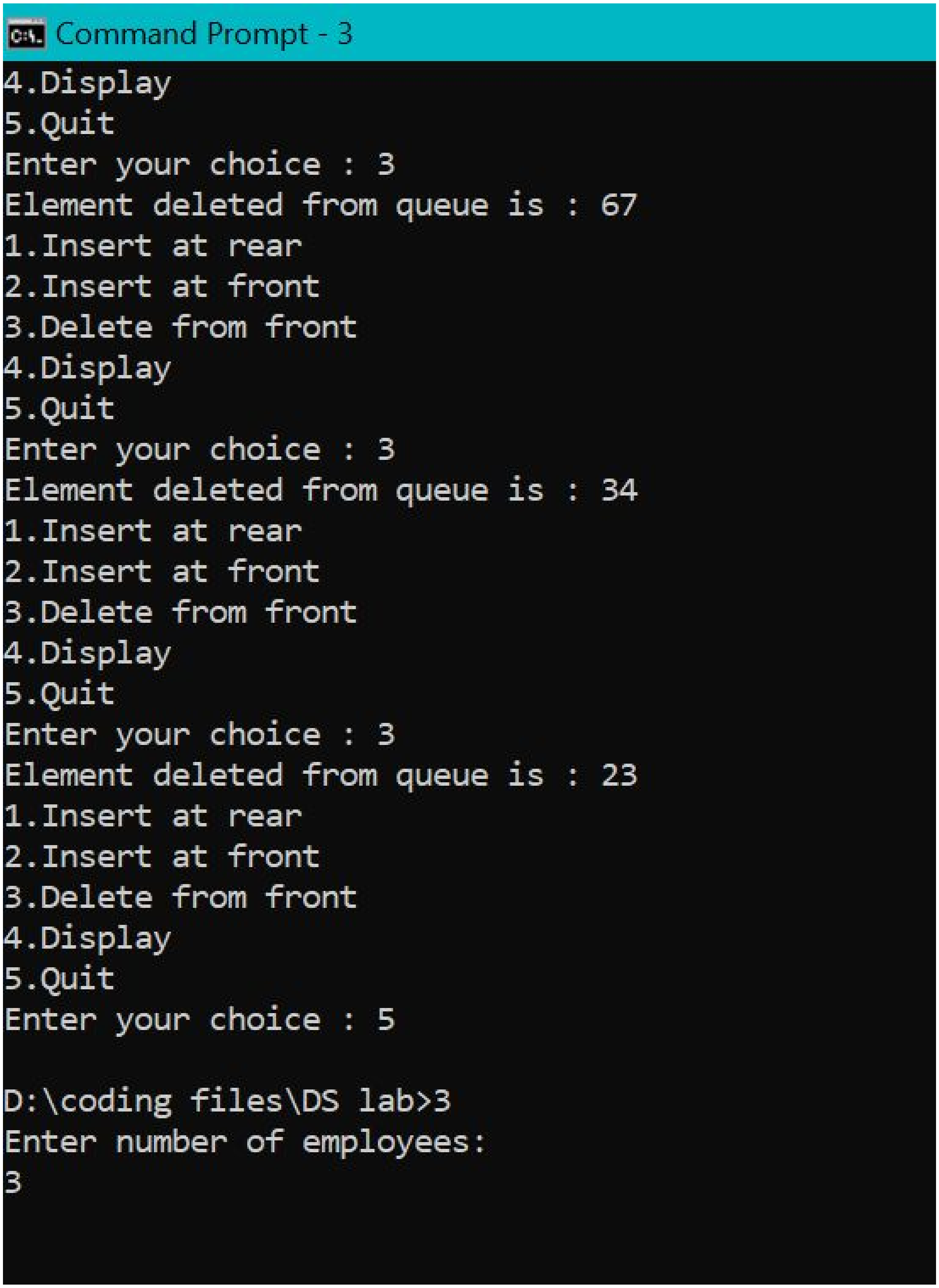
```
Enter the element for adding in queue : 67
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice :2
Element deleted from queue is : 23
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice :4
Queue elements :
34
45
67

1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice :3
Element deleted from queue is : 67
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice :5

D:\coding files\DS lab>1
Enter the total no of students
```



```
D:\coding files\DS lab>
D:\coding files\DS lab>restricted
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
Enter the element for adding in queue : 23
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 1
Enter the element for adding in queue : 45
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 2
Enter the element for adding in queue : 34
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 2
Enter the element for adding in queue : 67
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 2
Enter the element for adding in queue : 78
1.Insert at rear
2.Insert at front
3.Delete from front
```



```
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 1
***Queue Overflow***
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 4
Queue elements :
78
67
34
23
45

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 : 78
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 : 67
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 : 34
1.Insert at rear
2.Insert at front
```



```
4.Display
5.Quit
Enter your choice : 3
Element deleted from queue is : 67
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 : 34
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 : 23
1.Insert at rear
2.Insert at front
3.Delete from front
4.Display
5.Quit
Enter your choice : 5

D:\coding files\DS lab>3
Enter number of employees:
3
```