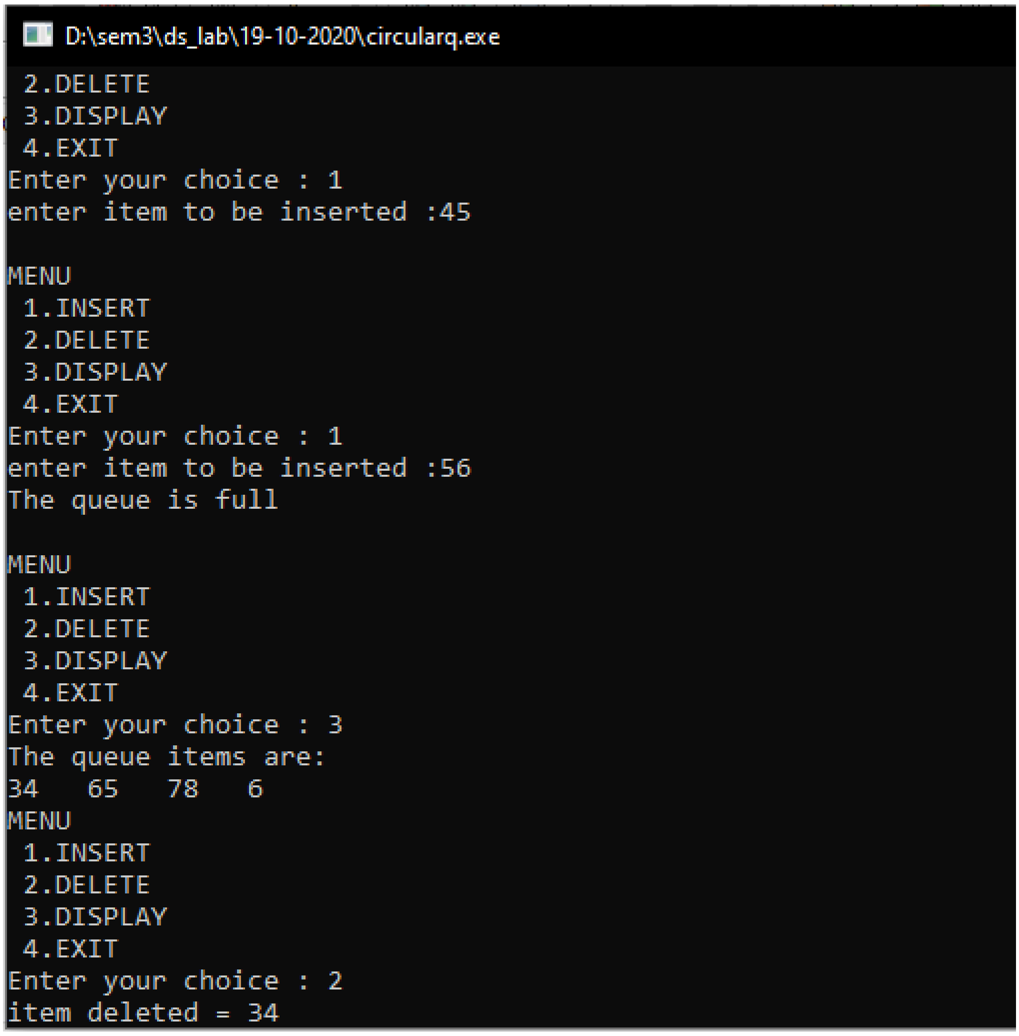
1. Circular queue implementation using arrays

```
#include<stdio.h>
#include<process.h>
# define queue_size 5
int item ,front=0, rear=-1, q[10],count=0;
void insert()
{
     if(count == queue_size)
     {
          printf("The queue is full \n");
          return ;
     }
     rear=(rear+1)%queue_size;
     q[rear]=item;
     count+=1;
}
int delete()
{
     if(count==0)
     {
          front=0;
          rear=-1;
          return -1;
     }
     item=q[front];
     front=(front+1)%queue_size;
     count-=1;
     return item;
}
void display()
{
     int i, f=front;
     if(count==0)
     {
          printf("The queue is empty");
          return;
     }
     printf("The queue items are: \n");
     for(i=1;i<+count;i++)
     {
          printf("%d   ",q[f]);
          f=(f+1)%queue_size;
```

```
    }
}
int main()
{
    int choice;
    for(;;)
    {
        printf("MENU \n 1.INSERT \n 2.DELETE \n 3.DISPLAY \n
        4.EXIT \n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch (choice)
        {
            case 1 :
                printf("enter item to be inserted :");
                scanf("%d",&item);
                insert();
                printf("item inserted\n");
                break;
            case 2 :
                item =delete();
                if(item==-1)
                    printf("the queue is empty \n");
                else
                    printf("item deleted = %d \n",item);
                break;
            case 3:
                display();
                break;
            default: exit(0);
        }
    }
    return 0;
}
```

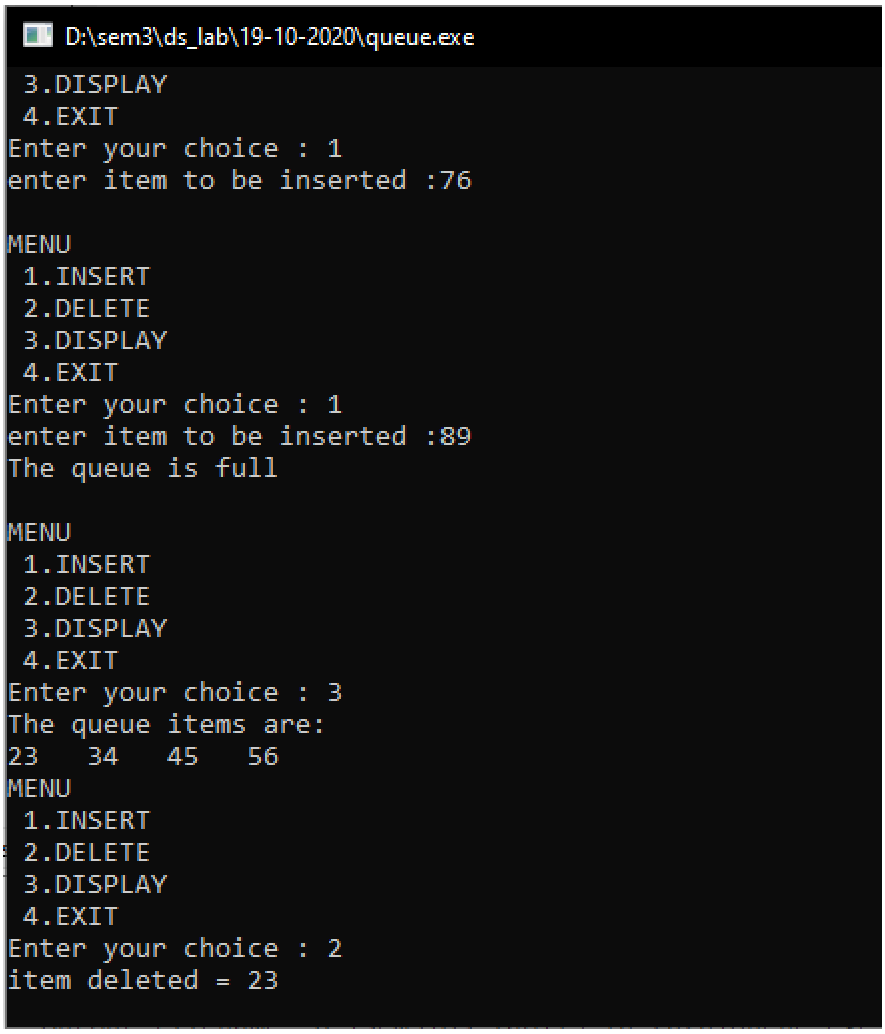


```
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter your choice : 1
enter item to be inserted :45

MENU
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter your choice : 1
enter item to be inserted :56
The queue is full

MENU
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter your choice : 3
The queue items are:
34    65    78    6
MENU
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter your choice : 2
item deleted = 34
```

2. Queue implementation using arrays

```
#include<stdio.h>
#include<process.h>
# define queue_size 5
int item ,front=0, rear=-1, q[10];
void insert()
{
	if(rear == queue_size-1)
	{
		printf("The queue is full \n");
		return ;
	}
	rear+=1;
	q[rear]=item;
}
int delete()
{
     if(front>rear)
     {
          front=0;
          rear=-1;
          return -1;
     }
     return q[front++];
}
void display()
{
     int i;
     if(rear>front)
     {
          printf("The queue is empty");
          return;
     }
     printf("The queue items are: \n");
     for(i=front;i<rear;i++)
     {
          printf("%d   ",q[i]);
     }
}
int main()
{
     int choice;
     for(;;)
     {
          printf("MENU \n 1.INSERT \n 2.DELETE \n 3.DISPLAY \n
4.EXIT \n");
          printf("Enter your choice : ");
          scanf("%d",&choice);
          switch (choice)
          {
               case 1 :
                    printf("enter item to be inserted :");
                    scanf("%d",&item);
                    insert();
                    printf("item inserted\n");
                    break;
               case 2 :
                    item =delete();
                    if(item==-1)
                         printf("the queue is empty \n");
```

```
                        else
                             printf("item deleted = %d \n",item);
                        break;
                   case 3:
                        display();
                        break;
                   default: exit(0);
              }
         }
         return 0;
    }
```



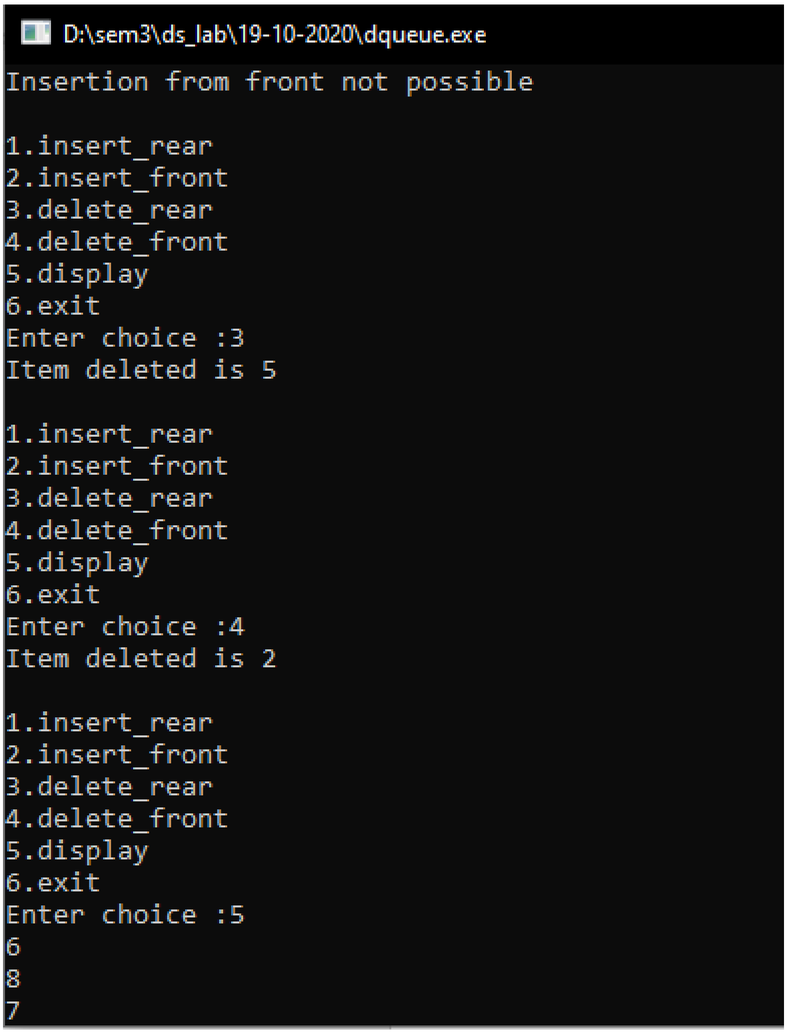
D:\sem3\ds_lab\19-10-2020\queue.exe

```
 3.DISPLAY
 4.EXIT
Enter your choice : 1
enter item to be inserted :76

MENU
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter your choice : 1
enter item to be inserted :89
The queue is full

MENU
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter your choice : 3
The queue items are:
23    34    45    56
MENU
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter your choice : 2
item deleted = 23
```

3. Double-ended queue implementation using arrays

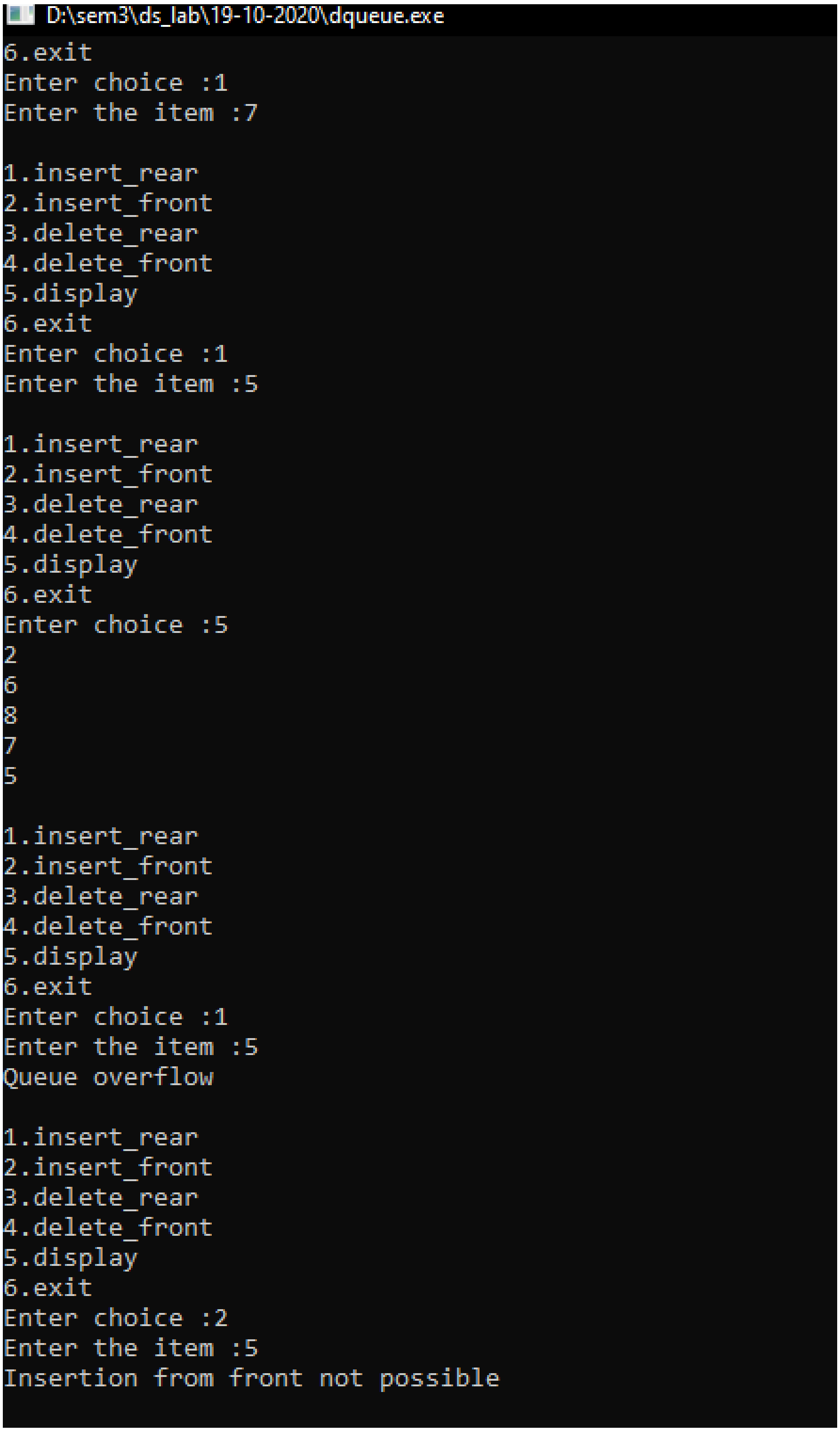
```
#include<stdio.h>
#include<process.h>
#define qsize 5
int f=0,r=-1,ch;
int item,q[10];

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("The queue is 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 from front not possible\n");
  }
void delete_rear()
  {
   if(isempty())
     {
      printf("The 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("The queue is empty\n");
      return;
     }
   for(i=f;i<=r;i++)
    printf("%d\n",q[i]);
  }
int main()
 {

  for(;;)
   {

printf("\n1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
    printf("Enter choice :");
    scanf("%d",&ch);
    switch(ch)
    {
       case 1:printf("Enter the item :");
              scanf("%d",&item);
              insert_rear();
              break;
       case 2:printf("Enter the item :");
              scanf("%d",&item);
              insert_front();
              break;
       case 3:delete_rear();
              break;
       case 4:delete_front();
              break;
       case 5:display();
              break;
       default:exit(0);
      }
    }
   return 0;
  }
```



D:\sem3\ds_lab\19-10-2020\dqueue.exe

```
Insertion from front not possible

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter choice :3
Item deleted is 5

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter choice :4
Item deleted is 2

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter choice :5
6
8
7
```



```
6.exit
Enter choice :1
Enter the item :7

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter choice :1
Enter the item :5

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter choice :5
2
6
8
7
5

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter choice :1
Enter the item :5
Queue overflow

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter choice :2
Enter the item :5
Insertion from front not possible
```

/\/\/\/\   /\/\/\/\   /\/\/\/\