# WEEK4-EXTRA LAB PROGRAMS

**PROGRAM-1**

**DOUBLE ENDED QUEUE**

```
#include<stdio.h>
#include<conio.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)--]);
```

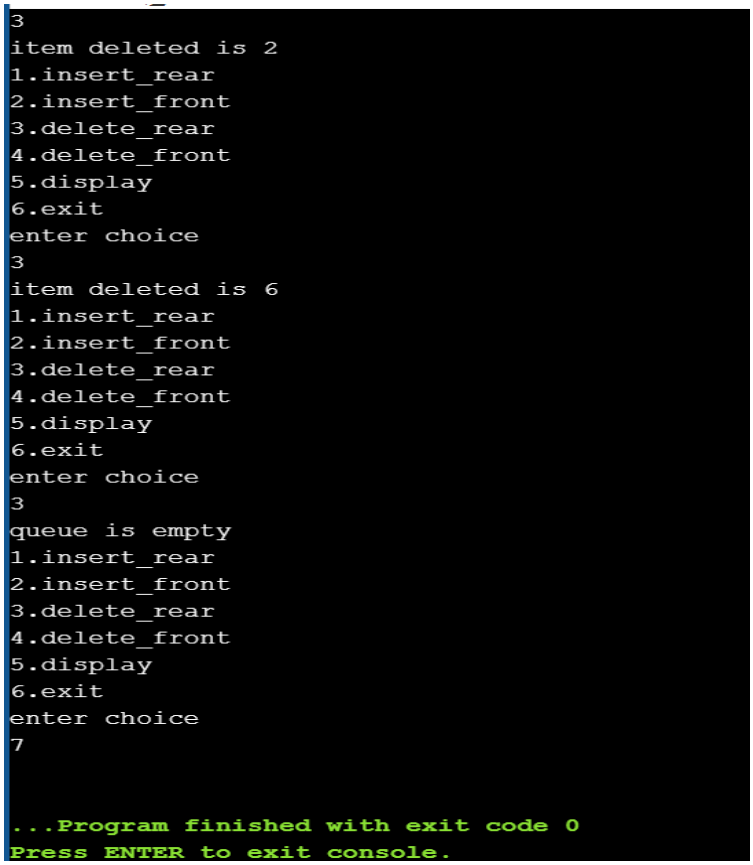
```
  if(f>r)
          {
           f=0;
           r=-1;
          }
 }
void display()
 {
  int i;
  if(isempty())
          {
           printf("queue empty\n");
           return;
          }
  for(i=f;i<=r;i++)
         printf("%d\n",q[i]);
 }
void main()
 {

 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);
         }
```

```
        }
 getch();
}
```

```
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
1
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
2
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
3
insertion not possible
1.insert_rear
```

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




## PROGRAM-2

## INPUT RESTRICTED

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define qsize 3

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 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;
```

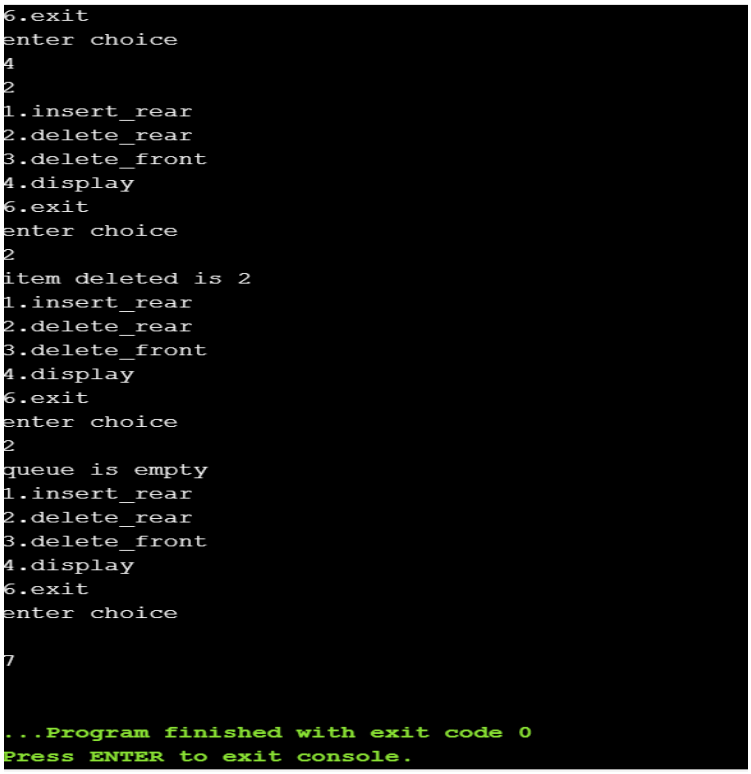
```
 if(isempty())
        {
         printf("queue empty\n");
         return;
        }
 for(i=f;i<=r;i++)
        printf("%d\n",q[i]);
 }
void main()
{

 for(;;)
 {
        printf("1.insert_rear\n2.delete_rear\n3.delete_front\n4.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:delete_rear();
                        break;
```

```
            case 3:delete_front();
                                break;
            case 4:display();
                                break;
            default:exit(0);
          }
        }
        getch();
}
```

```
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
1
enter the item
1
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
1
enter the item
2
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
1
enter the item
3
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
```

```
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
4
1
2
3
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
1
enter the item
4
queue overflow
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
2
item deleted is 3
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
```

```
3.delete_front
4.display
6.exit
enter choice
1
enter the item
4
queue overflow
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
2
item deleted is 3
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
3
item deleted is 1
1.insert_rear
2.delete_rear
3.delete_front
4.display
6.exit
enter choice
4
2
```




**PROGRAM-3**

**OUTPUT RESTRICTED**

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define qsize 3

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 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_front\n4.display\n6.exit\
n");
        printf("enter choice\n");
        scanf("%d",&ch);
        switch(ch)
```

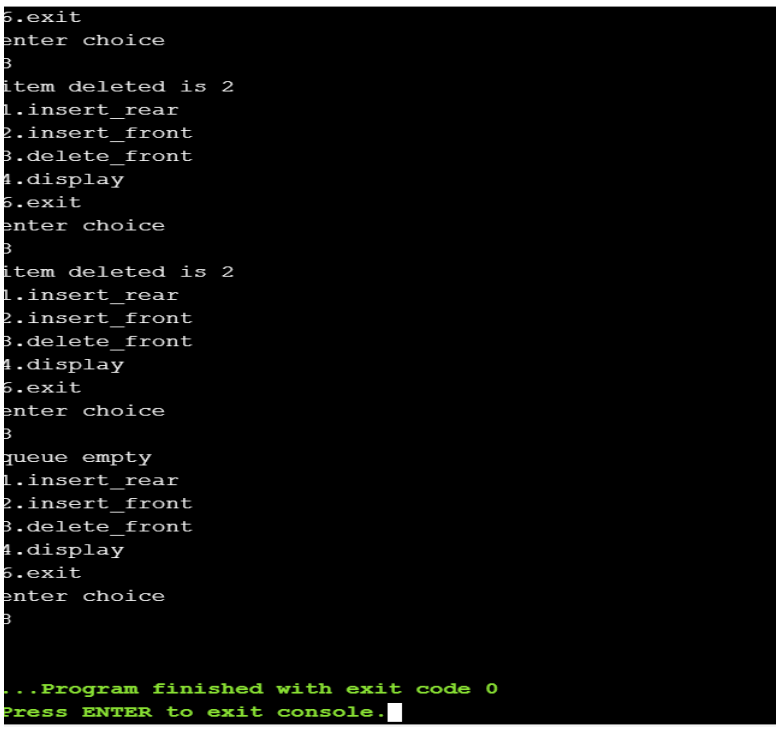
```
        {
         case 1:printf("enter the item\n");
                        scanf("%d",&item);
                        insert_rear();
                        break;
         case 2:printf("enter the item\n");
                        scanf("%d",&item);
                        insert_front();
                        break;
         case 3:delete_front();
                        break;
         case 4:display();
                        break;
         default:exit(0);
        }
       }
       getch();
}
```

```
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
1
enter the item
1
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
2
enter the item
2
insertion not possible
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
1
enter the item
2
1.insert_rear
2.insert_front
3.delete_front
```

```
4.display
6.exit
enter choice
1
enter the item
2
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
1
enter the item
3
queue overflow
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
4
1
2
2
1.insert_rear
2.insert_front
3.delete_front
4.display
```

```
enter choice
3
item deleted is 1
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
2
enter the item
4
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
4
4
2
2
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
1
enter the item
5
```

```
queue overflow
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
3
item deleted is 4
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
3
item deleted is 2
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
3
item deleted is 2
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
```


6.exit
enter choice
3
item deleted is 2
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
3
item deleted is 2
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
3
queue empty
1.insert_rear
2.insert_front
3.delete_front
4.display
6.exit
enter choice
3
...Program finished with exit code 0
Press ENTER to exit console.