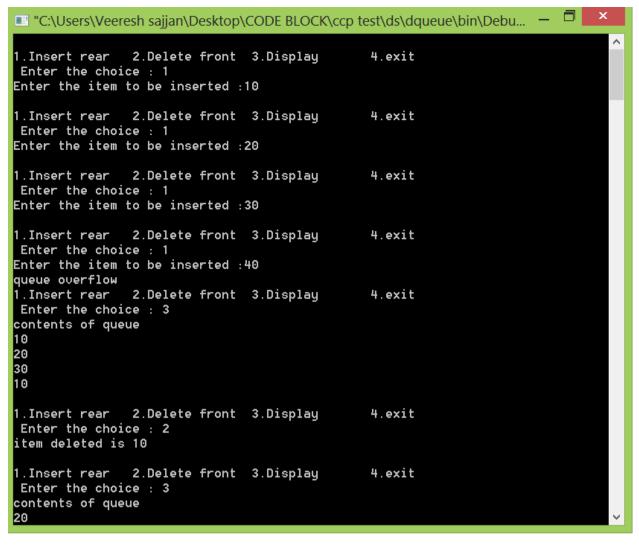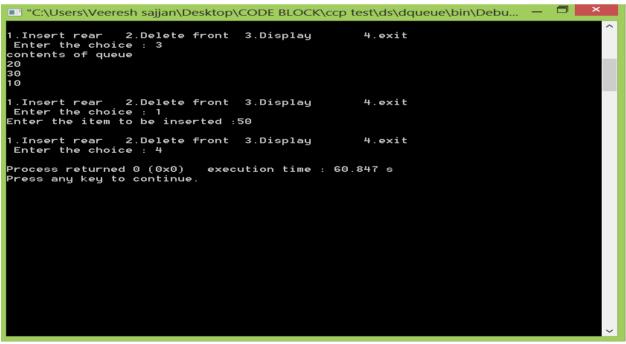# 1) Circular Queue:

```c
#include<stdio.h>

#include<stdlib.h>

#include<process.h>

#define que_size 3

int item,front=0,rear=-1,q[que_size],count=0;

void insertrear()
{
    if(count==que_size)
    {
        printf("queue overflow");
        return;
    }
    rear=(rear+1)%que_size;
    q[rear]=item;
    count++;
}
int deletefront()
{
    if(count==0) return -1;
    item = q[front];
    front=(front+1)%que_size;
    count=count-1;
    return item;
}
```

```c
void displayq()
{
    int i,f;
    if(count==0)
    {
        printf("queue is empty");
        return;
    }
    f=front;
    printf("contents of queue \n");
    for(i=0;i<=count;i++)
    {
        printf("%d\n",q[f]);
        f=(f+1)%que_size;
    }
}
void main()
{
    int choice;
    for(;;)
    {
        printf("\n1.Insert rear \t2.Delete front \t3.Display \t4.exit \n ");
        printf("Enter the choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
```

```c
            case 1:printf("Enter the item to be inserted :");

                   scanf("%d",&item);

                   insertrear();

                   break;

            case 2:item=deletefront();

                       if(item==-1)

                       printf("queue is empty\n");

                       else

                       printf("item deleted is %d \n",item);

                       break;

         case 3:displayq();

                       break;

         default:exit(0);

        }

    }

}
```
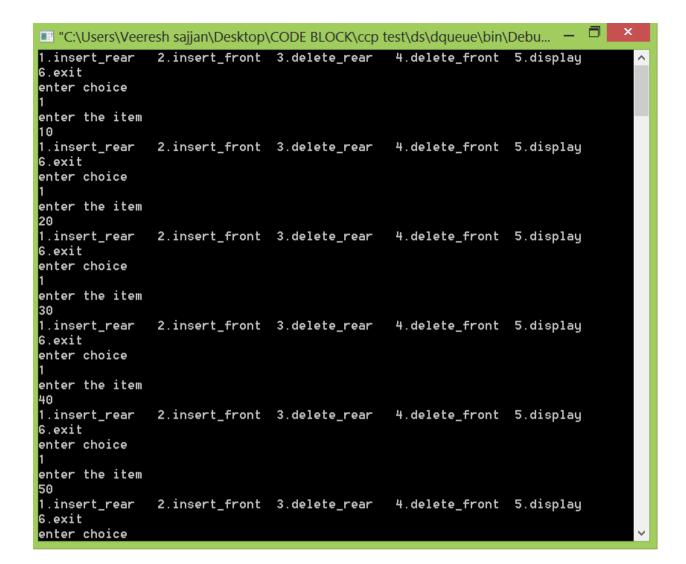
```
"C:\Users\Veeresh sajjan\Desktop\CODE BLOCK\ccp test\ds\dqueue\bin\Debu...   —  □  ✕

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 1
Enter the item to be inserted :10

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 1
Enter the item to be inserted :20

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 1
Enter the item to be inserted :30

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 1
Enter the item to be inserted :40
queue overflow
1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 3
contents of queue
10
20
30
10

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 2
item deleted is 10

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 3
contents of queue
20
```

```
"C:\Users\Veeresh sajjan\Desktop\CODE BLOCK\ccp test\ds\dqueue\bin\Debu...   —  □  ✕

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 3
contents of queue
20
30
10

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 1
Enter the item to be inserted :50

1.Insert rear    2.Delete front  3.Display      4.exit
 Enter the choice : 4

Process returned 0 (0x0)   execution time : 60.847 s
Press any key to continue.
```

2) D-queue:

```c
#include<stdio.h>

#include<conio.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())
```

```c
            {
             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");
```

```c
        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()
    {

    clrscr();
    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");
```

```c
                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
10
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
1
enter the item
20
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
1
enter the item
40
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
1
enter the item
50
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
```

```
enter the item
50
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
5
10
20
30
40
50
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
1
enter the item
60
queue overflow
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
3
item deleted is 50
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear    2.insert_front  3.delete_rear    4.delete_front  5.display
6.exit
enter choice
2
enter the item
```

```
enter choice
4
item deleted is 10
1.insert_rear    2.insert_front   3.delete_rear    4.delete_front   5.display
6.exit
enter choice
2
enter the item
60
1.insert_rear    2.insert_front   3.delete_rear    4.delete_front   5.display
6.exit
enter choice
2
enter the item
70
insertion not possible
1.insert_rear    2.insert_front   3.delete_rear    4.delete_front   5.display
6.exit
enter choice
5
60
20
30
40
1.insert_rear    2.insert_front   3.delete_rear    4.delete_front   5.display
6.exit
enter choice
```