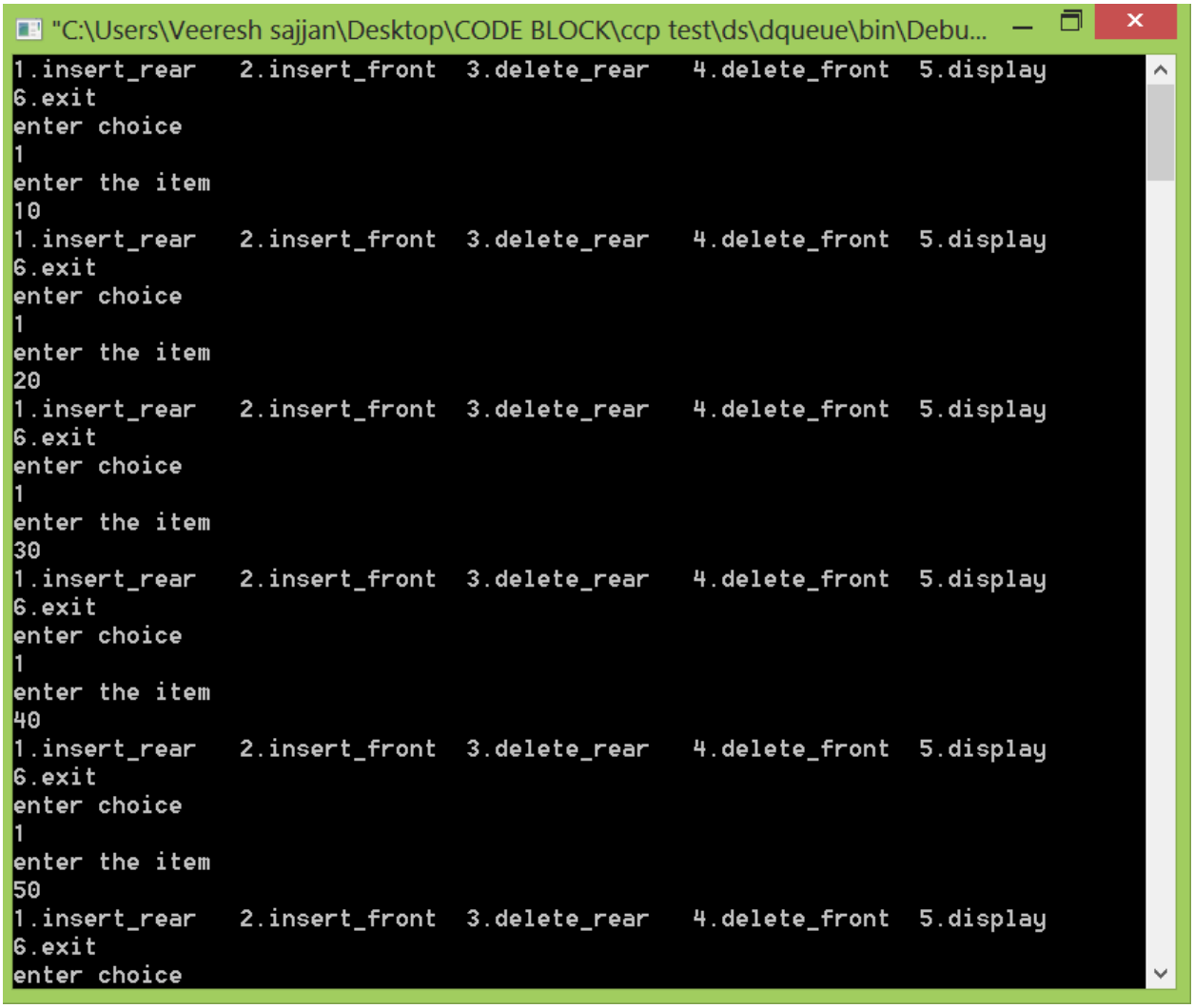
2) D-queue:

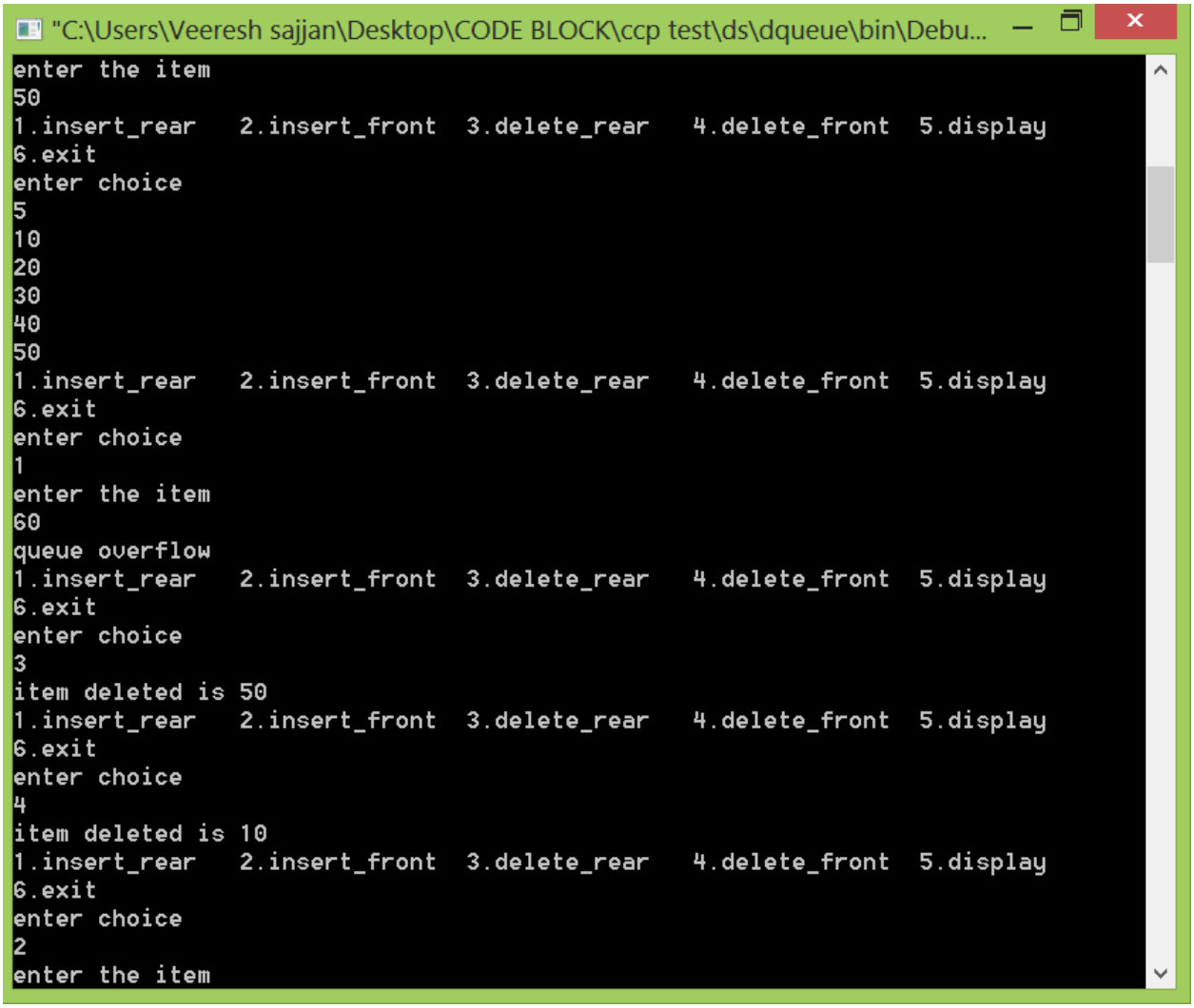
```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#define qsize 5
int f=0,r=-1,ch;
int item,q[10];

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())
          {
           printf("queue empty\n");
```

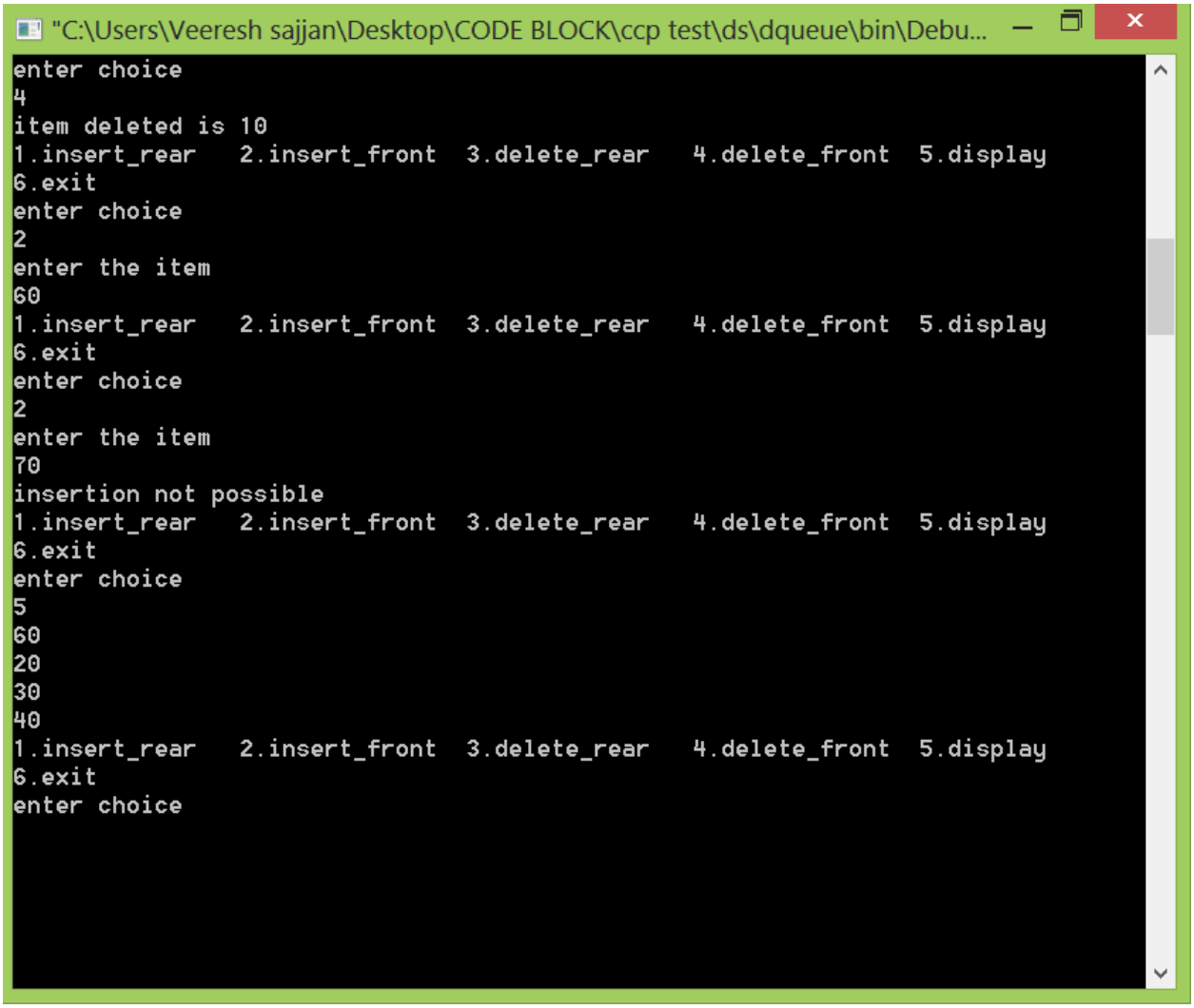
```
            return;
           }
    printf("item deleted is %d\n",q[(f)++]);
    if(f>r)
           {
            f=0;
            r=-1;
           }
   }
void insert_front()
   {
    if(f!=0)
           {
            f=f-1;
            q[f]=item;
            return;
           }
          else 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()
 {

  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");
                          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();
}
```


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


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


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