main.c

```c
//DOUBLE ENDED QUEUE PROGRAM

#include<stdio.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;
      f=0;
      r=-1;
     }
   printf("Item deleted is %d\n",q[(f)++]);
  }
void insert_front()
  {
```

main.c

```
37    }
38  void insert_front()
39    {
40     if(f!=0)
41       {
42        f=f-1;
43        q[f]=item;
44        return;
45       }
46      else if((f==0)&&(r==-1))
47       {
48        q[++(r)]=item;
49        return;
50       }
51      else
52        printf("Insertion not possible\n");
53    }
54  void delete_rear()
55    {
56     if(isempty())
57       {
58        printf("Queue is empty\n");
59        return;
60        f=0;
61        r=-1;
62       }
63     printf("Item deleted is %d\n",q[(r)--]);
64    }
65  void display()
66    {
67     int i;
68     if(isempty())
69       {
70        printf("Queue empty\n");
71        return;
72       }
73     for(i=f;i<=r;i++)
74      printf("%d\n",q[i]);
75    }
```

main.c

```c
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);
      }
    }
  }
```

```
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
```

```
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
Queue overflow
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
5
10
20
30
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
3
Item deleted is 30
1.Insert_rear
2.Insert_front
```

```
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
2
Enter the item
60
Insertion not possible
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
5
10
20
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
5
20
```

main.c

```c
//INPUT RESTRICTED DEQUEUE

#include<stdio.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;
      f=0;
      r=-1;
     }
   printf("Item deleted is %d\n",q[(f)++]);
```

main.c

```
37      }
38     printf("Item deleted is %d\n",q[(f)++]);
39    }
40  void delete_rear()
41    {
42     if(isempty())
43       {
44        printf("Queue is empty\n");
45        return;
46        f=0;
47        r=-1;
48       }
49     printf("Item deleted is %d\n",q[(r)--]);
50    }
51
52
53  void display()
54    {
55     int i;
56     if(isempty())
57       {
58        printf("Queue empty\n");
59        return;
60       }
61     for(i=f;i<=r;i++)
62      printf("%d\n",q[i]);
63    }
64
65
66  void main()
67   {
68
69    for(;;)
70     {
71      printf("1.Insert_rear\n2.Delete_rear\n3.Delete_front\n4.Display\n5.Exit\n");
72      printf("Enter choice\n");
73      scanf("%d",&ch);
74      switch(ch)
75        {
```

main.c

```
53  void display()
54    {
55     int i;
56     if(isempty())
57       {
58        printf("Queue empty\n");
59        return;
60       }
61     for(i=f;i<=r;i++)
62      printf("%d\n",q[i]);
63    }
64
65
66  void main()
67   {
68
69    for(;;)
70     {
71      printf("1.Insert_rear\n2.Delete_rear\n3.Delete_front\n4.Display\n5.Exit\n");
72      printf("Enter choice\n");
73      scanf("%d",&ch);
74      switch(ch)
75        {
76         case 1:printf("Enter the item\n");
77                scanf("%d",&item);
78                insert_rear();
79                break;
80         case 2:delete_rear();
81                break;
82         case 3:delete_front();
83                break;
84         case 4:display();
85                break;
86         default:exit(0);
87        }
88      }
89    }
90
```

input

```
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
20
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
40
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
60
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
30
Queue overflow
```

```
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
4
20
40
60
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
2
Item deleted is 60
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
4
20
40
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
2
Item deleted is 40
1.Insert_rear
2.Delete_rear
```

```
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
4
20
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
90
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
4
20
90
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
5


...Program finished with exit code 0
Press ENTER to exit console.
```

main.c

```c
//OUTPUT RESTRICTED DEQUEUE

#include<stdio.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;
      f=0;
      r=-1;
     }
```

main.c

```
37        f=0;
38        r=-1;
39       }
40     printf("Item deleted is %d\n",q[(f)++]);
41    }
42
43
44  void insert_front()
45    {
46     if(f!=0)
47       {
48        f=f-1;
49        q[f]=item;
50        return;
51       }
52      else if((f==0)&&(r==-1))
53       {
54        q[++(r)]=item;
55        return;
56       }
57      else
58        printf("Insertion not possible\n");
59    }
60  void display()
61    {
62     int i;
63     if(isempty())
64       {
65        printf("Queue empty\n");
66        return;
67       }
68     for(i=f;i<=r;i++)
69      printf("%d\n",q[i]);
70    }
71
72
73  void main()
74   {
75
```

main.c

```
61   {
62    int i;
63    if(isempty())
64      {
65       printf("Queue empty\n");
66       return;
67      }
68    for(i=f;i<=r;i++)
69     printf("%d\n",q[i]);
70   }
71
72
73  void main()
74   {
75
76    for(;;)
77     {
78      printf("1.Insert_rear\n2.Insert_front\n3.Delete_front\n4.Display\n5.Exit\n");
79      printf("Enter choice\n");
80      scanf("%d",&ch);
81      switch(ch)
82        {
83         case 1:printf("Enter the item\n");
84                scanf("%d",&item);
85                insert_rear();
86                break;
87         case 2:printf("Enter the item\n");
88                scanf("%d",&item);
89                insert_front();
90                break;
91         case 3:delete_front();
92                break;
93         case 4:display();
94                break;
95         default:exit(0);
96        }
97      }
98   }
99
```

```
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
15
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
25
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
35
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
45
Queue overflow
```

```
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 15
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
4
25
35
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
2
Enter the item
12
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
2
Enter the item
99
Insertion not possible
1.Insert_rear
```

```
Enter choice
4
12
25
35
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 12
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 25
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 35
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Queue empty
```