

## LAB-4 DOUBLE ENDED QUEUE

```
# include < stdio.h >
# define qsize 5
# include < stdlib.h >
# define qsize 5
int f=0, r=-1, ch;
int item, q[10];
int is full()
{
    return (r == qsize - 1) ? 1 : 0;
}
int is empty()
{
    return (f > r) ? 1 : 0;
}
void insert_rear()
{
    if (is full())
        printf (" queue overflow \n");
    return;
}
```

$n = r + 1$ ,  
 $q[n] = \text{item}$ ,  
}

void deleteFront()

{

if ( $f == n$ )

printf ("queue empty\n");  
return;

}

printf ("item deleted is %d\n", q[f++]);

if ( $f > n$ )

{

$f = 0$ ;

$n = -1$ ;

}

}

void insertFront()

{

if ( $f == 0$ )

$f = f - 1$

$q[f] = \text{item}$ ;

return;

}

```
else if (f == 0) && (r == -1)
```

{

```
q[ ++(r) ] = item;
```

```
return,
```

}

else

```
printf ("insertion not possible\n");
```

},

```
void delete_rear()
```

{

```
if (is_empty())
```

```
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 (is empty())
{
    printf ("queue empty \n");
    return;
}

for(i=1; i<=n; i++)
    printf ("%d", q[i]);
}
```

```
void main()
for(;;)
```

printf ("1.insert-rear\n 2. insert-front\n 3.  
delete-rear\n 4. delete-front\n 5.display  
6. 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);

}

4

3

```
C:\Users\Danish\Downloads\file\Wlinkdque
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
1
enter the item
3
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
4
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 4
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
1
2
3
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
6
Process returned 0 (0x0) execution time : 130.711 s
Press any key to continue.
```

SVA\_DQA (4) - Notepad

```
#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;
```

SVA\_DQA (4) - Notepad

```
File Edit Format View Help
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;
    }
}
```

08:18 PM  
08-11-2020

SVA\_DQA (4) - Notepad

```
File Edit Format View Help
}
    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();
        }
    }
}
```

SVA\_DQA (4) - Notepad

```
File Edit Format View Help
        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();
}
```

08-11-2020 08:19 PM

Upload files · revanths128/DS-LA x | Data-Structure-lab/INPUT RESTRICTED DQUEUE.pdf x | INPUT RESTRICTED DQUEUE.pdf x | Repl.it - C Online Compiler, IDE, | +

← → C 🔒 repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define qsize 3
4 int f=0,r=-1,ch;
5 int item,q[10];
6 int isfull()
7 {
8     return(r==qsize-1)?1:0;
9 }
10 int isempty()
11 {
12     return(f>r)?1:0;
13 }
14 void insert_rear()
15 {
16     if(isfull())
17     {
18         printf("queue overflow\n");
19         return;
20     }
21     r=r+1;
22     q[r]=item;
23 }
24 void delete_front()
25 {
26     if(isempty())
27     {
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 2
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
1
2
3
4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
```

Upload files · revanths128/DS-LA | Data-Structure-lab/INPUT RESTRICTIONS | INPUT RESTRICTED DQUEUE.pdf | Repl.it - C Online Compiler, IDE, | +

← → C repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
22 q[r]=item;
23 }
24 void delete_front()
25 {
26 if(isempty())
27 {
28 printf("queue empty\n");
29 return;
30 }
31 printf("item deleted is %d\n",q[(f++)]);
32 if(f>r)
33 {
34 f=0;
35 r=-1;
36 }
37 }
38 void delete_rear()
39 {
40 if(isempty())
41 {
42 printf("queue is empty\n");
43 return;
44 }
45 printf("item deleted is %d\n",q[(r)--]);
46 if(f>r)
47 {
48 f=0;
49 r=1;
50 }
51 }
```

```
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
1
2
3
4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 2
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
queue is empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
queue empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 5
```

Upload files · revanths128/DS-LA | Data-Structure-lab/INPUT RESTRICTIONS | INPUT RESTRICTED DQUEUE.pdf | Repl.it - C Online Compiler, IDE, | +

← → C repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
47 {
48     f=0;
49     r=-1;
50 }
51
52 void display()
53 {
54     int i;
55     if(isempty())
56     {
57         printf("queue empty\n");
58         return;
59     }
60     for(i=f;i<=r;i++)
61     printf("%d\n",q[i]);
62 }
63 int main()
64 {
65     for(;;)
66     {
67         printf("1.insert_rear 2.delete_rear 3.delete_front 4.display
68         5.exit\n");
69         printf("enter choice : ");
70         scanf("%d",&ch);
71         switch(ch)
72         {
73             case 1:printf("enter the item : ");
74             break;
75             case 2:if(r>=f)
76                 printf("queue empty\n");
77             else
78                 {
79                     printf("item deleted is %d\n",q[f]);
80                     f++;
81                 }
82             break;
83             case 3:if(f>=r)
84                 printf("queue empty\n");
85             else
86                 {
87                     printf("item deleted is %d\n",q[r]);
88                     r--;
89                 }
90             break;
91             case 4:display();
92             break;
93             case 5:exit(0);
94         }
95     }
96 }
```

1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 1  
enter the item : 4  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 4  
1  
2  
3  
4  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 2  
item deleted is 4  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 2  
item deleted is 3  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 3  
item deleted is 1  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 3  
item deleted is 2  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 2  
queue is empty  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 4  
queue empty  
1.insert\_rear 2.delete\_rear 3.delete\_front 4.display 5.exit  
enter choice : 5

Upload files · revanths128/DS-LA | Data-Structure-lab/INPUT RESTRICTED | INPUT RESTRICTED DQUEUE.pdf | Repl.it - C Online Compiler, IDE, | +

← → C repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
1
2
3
4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 4
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
item deleted is 3
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 2
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 2
queue is empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 4
queue empty
1.insert_rear 2.delete_rear 3.delete_front 4.display 5.exit
enter choice : 5
```

Upload files · revanths128/DS-LA x | Data-Structure-lab/OUTPUT RES x | OUTPUT RESTRICTED DQUEUE.p x | Repl.it - C Online Compiler, IDE, | +

← → C 🔒 repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define qsize 3
4 int f=0,r=-1,ch;
5 int item,q[10];
6 int isfull()
7 {
8     return(r==qsize-1)?1:0;
9 }
10 int isempty()
11 {
12     return(f>r)?1:0;
13 }
14 void insert_rear()
15 {
16     if(isfull())
17     {
18         printf("queue overflow\n");
19         return;
20     }
21     r=r+1;
22     q[r]=item;
23 }
24 void delete_front()
25 {
26     if(isempty())
27     {
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 2
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 4
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 4
1
3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
queue empty
```

Upload files · revanths128/DS-LA x | Data-Structure-lab/OUTPUT RES x | OUTPUT RESTRICTED DQUEUE.p x | Repl.it - C Online Compiler, IDE, | +

← → C 🔒 repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
-- 26     if(isempty())
27 {
28     printf("queue empty\n");
29     return;
30 }
31 printf("item deleted is %d\n",q[(f++)];
32     if(f>r)
33 {
34     f=0;
35     r=-1;
36 }
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");
```

```
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 2
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 4
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 4
1
3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
queue empty
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 5
```

Upload files · revanths128/DS-LA x | Data-Structure-lab/OUTPUT RES x | OUTPUT RESTRICTED DQUEUE.p x | Repl.it - C Online Compiler, IDE, | +

← → C 🔒 repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
50 }  
51 else  
52 printf("insertion not possible\n");  
53 }  
54 void display()  
55 {  
56     int i;  
57     if(isempty())  
58     {  
59         printf("queue empty\n");  
60         return;  
61     }  
62     for(i=f;i<=r;i++)  
63     printf("%d\n",q[i]);  
64 }  
65 int main()  
66 {  
67     for(;;)  
68     {  
69         printf("1.insert_rear 2.insert_front 3.delete_front 4.display  
5.exit\n");  
70         printf("enter choice : ");  
71         scanf("%d",&ch);  
72         switch(ch)  
73     {  
74         case 1:printf("enter the item : ");  
75         scanf("%d",&item);  
76         insert_rear();  
77     }  
78     }  
79 }  
80 }
```

1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 1  
enter the item : 1  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 2  
enter the item : 2  
insertion not possible  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 1  
enter the item : 3  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 2  
enter the item : 4  
insertion not possible  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 4  
1  
3  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 3  
item deleted is 1  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 3  
item deleted is 3  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 3  
queue empty  
1.insert\_rear 2.insert\_front 3.delete\_front 4.display 5.exit  
enter choice : 5

Upload files · revanths128/DS-LA x | Data-Structure-lab/OUTPUT RES x | OUTPUT RESTRICTED DQUEUE.p x | Repl.it - C Online Compiler, IDE, | +

← → C 🔒 repl.it/languages/c

Apps Gmail YouTube Maps

repl.it Features Jobs Blog Pricing Jam Log in Sign up

C C Run ▶ Share

```
68+ {
69 printf("1.insert_rear 2.insert_front 3.delete_front 4.display
      5.exit\n");
70 printf("enter choice : ");
71 scanf("%d",&ch);
72 switch(ch)
73+ {
74 case 1:printf("enter the item : ");
75 scanf("%d",&item);
76 insert_rear();
77 break;
78 case 2:printf("enter the item : ");
79 scanf("%d",&item);
80 insert_front();
81 break;
82 case 3:delete_front();
83 break;
84 case 4:display();
85 break;
86 default:exit(0);
87
88 }
89 }
90
91 }
```

```
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 2
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 1
enter the item : 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 2
enter the item : 4
insertion not possible
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 4
1
3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 1
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
item deleted is 3
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 3
queue empty
1.insert_rear 2.insert_front 3.delete_front 4.display 5.exit
enter choice : 5
```