

DS LAB-Practice progs- Dequeue, Input and output restricted dequeue

Program and output

Mallika Prasad

1BM19CS081

28.10.2020

Program-1

Implement dequeue-

```
#include<stdio.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]);
}
int main()
{

do
{
    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:break;

}

}

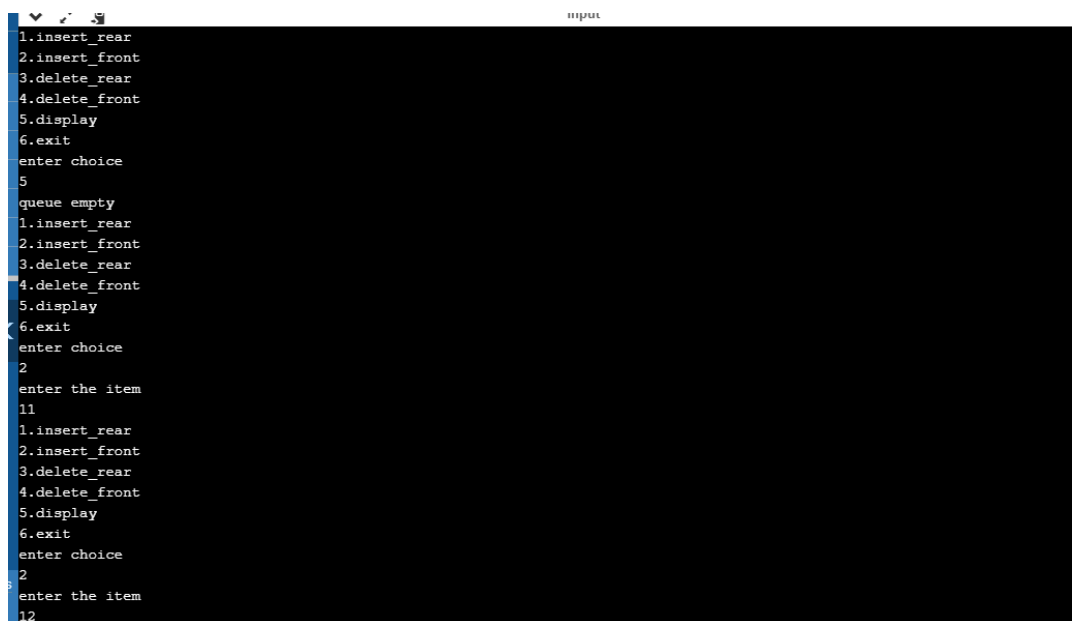
while(ch!=6);

return 0;

}

```

Output-



```

input
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
queue empty
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
11
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
4.delete_front
5.display
6.exit
enter choice
4
enter the item
12
5.display
6.exit
enter choice
5
6.exit
enter choice
6
enter the item
12

```

```
12
insertion not possible
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
12
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
13
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
```

```
1
enter the item
14
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
15
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
16
queue overflow
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
```

```
5.display
6.exit
enter choice
5
11
12
13
14
15
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 15
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 14
1.insert_rear
2.insert_front
```

```
- - -
item deleted is 14
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
11
12
13
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
21
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```

```
1
enter the item
43
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
11
12
13
21
43
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 11
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
```

```
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 12
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
13
21
43
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
55
1.insert_rear
```



```
6.exit
enter choice
2
enter the item
55
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
55
13
21
43
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
6

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

Program 2-

Input restricted dequeue

```
#include<stdio.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 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]);
}

int main()
```

```

{
printf("1.insert_rear\n2.delete_rear\n3.delete_front\n4.display\n5.exit\n");
    do{
        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:break;
        }
    }
while(ch!=5);
return 0;
}

```

Output-

```
1.insert_rear
2.delete_rear
3.delete_front
4.display
5.exit
enter choice
1
enter the item
11
enter choice
1
enter the item
12
enter choice
1
enter the item
13
enter choice
1
enter the item
14
enter choice
1
enter the item
15
enter choice
1
enter the item
16
queue overflow
```

```
15
enter choice
1
enter the item
16
queue overflow
enter choice
4
11
12
13
14
15
enter choice
2
item deleted is 15
enter choice
3
item deleted is 11
enter choice
4
12
13
14
enter choice
5

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

Program 3-

Output restricted dequeue

```
#include<stdio.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 display()
{
    int i;
    if(isempty())
    {
        printf("queue empty\n");
        return;
    }
    for(i=f;i<=r;i++)
        printf("%d\n",q[i]);
}
int main()
{
    printf("1.insert_rear\n2.insert_front\n3.delete_front\n4.display\n5.exit\n");
    do{
        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:break;

    }

}

while(ch!=5);

return 0;

}

```

Output-

```

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
enter choice
2
enter the item
11
enter choice
2
enter the item
12
insertion not possible
enter choice
1
enter the item
12
enter choice
1
enter the item
13
enter choice
1
enter the item
14
enter choice
1
enter the item
15
enter choice

```

```
enter choice
1
enter the item
16
queue overflow
enter choice
4
11
12
13
14
15
enter choice
3
item deleted is 11
enter choice
3
item deleted is 12
enter choice
4
13
14
15
enter choice
1
enter the item
22
queue overflow
enter choice
2
enter the item
```

```
enter choice
2
enter the item
22
enter choice
4
22
13
14
15
enter choice
3
item deleted is 22
enter choice
3
item deleted is 13
enter choice
3
item deleted is 14
enter choice
3
item deleted is 15
enter choice
4
queue empty
enter choice
5

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