

**Rajshahi University of Engineering & Technology**  
**Department of Computer Science of Engineering**

**EXPERIMENT NO: 07**  
**NAME OF EXPERIMENT: Queue**

**SUBMITTED TO:**

**RIZOAN TOUFIQ**  
**ASSISTANT PROFESSOR**  
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**SUBMITTED BY:**

**NAME: MD. ARIFUL ISLAM**  
**ROLL No.: 1803046**  
**GROUP: 2ND THIRTY**  
**DATE OF EXP.: 09-11-2019**  
**DATE OF SUB. : 17-11-2019**  
**SERIES: 18 SERIES**

**MACHINE CONFIGURATION:**

**ASUS X510UF**  
**CORE I5 8TH GEN PROCESSOR**  
**UP To 3.4 GHz**  
**8 GB RAM**  
**OS WIN 10**

INDEX

<b>Preblem No.</b>	<b>Name of Problem</b>	<b>Page No.</b>
<b>01.</b>	Inserting an item into a deque at right end	<b>04-05</b>
<b>02.</b>	Inserting an item into a deque at left end	<b>06-07</b>
<b>03.</b>	Deleting an item into a deque at right end	<b>08-09</b>
<b>04.</b>	Deleting an item into a deque at left end	<b>10-11</b>
<b>05.</b>	<empty>	
<b>06.</b>	<empty>	
<b>07.</b>	<empty>	
<b>08</b>	<empty>	
<b>09</b>	<empty>	
<b>10.</b>	<empty>	

**THEORY:** A Queue is a linear structure in which items may be added one by one only at the end or removed one by one only at the beginning. It means that the last item to be added to a queue is the last item to be removed.

The main concept of queue is **First In - First Out**.

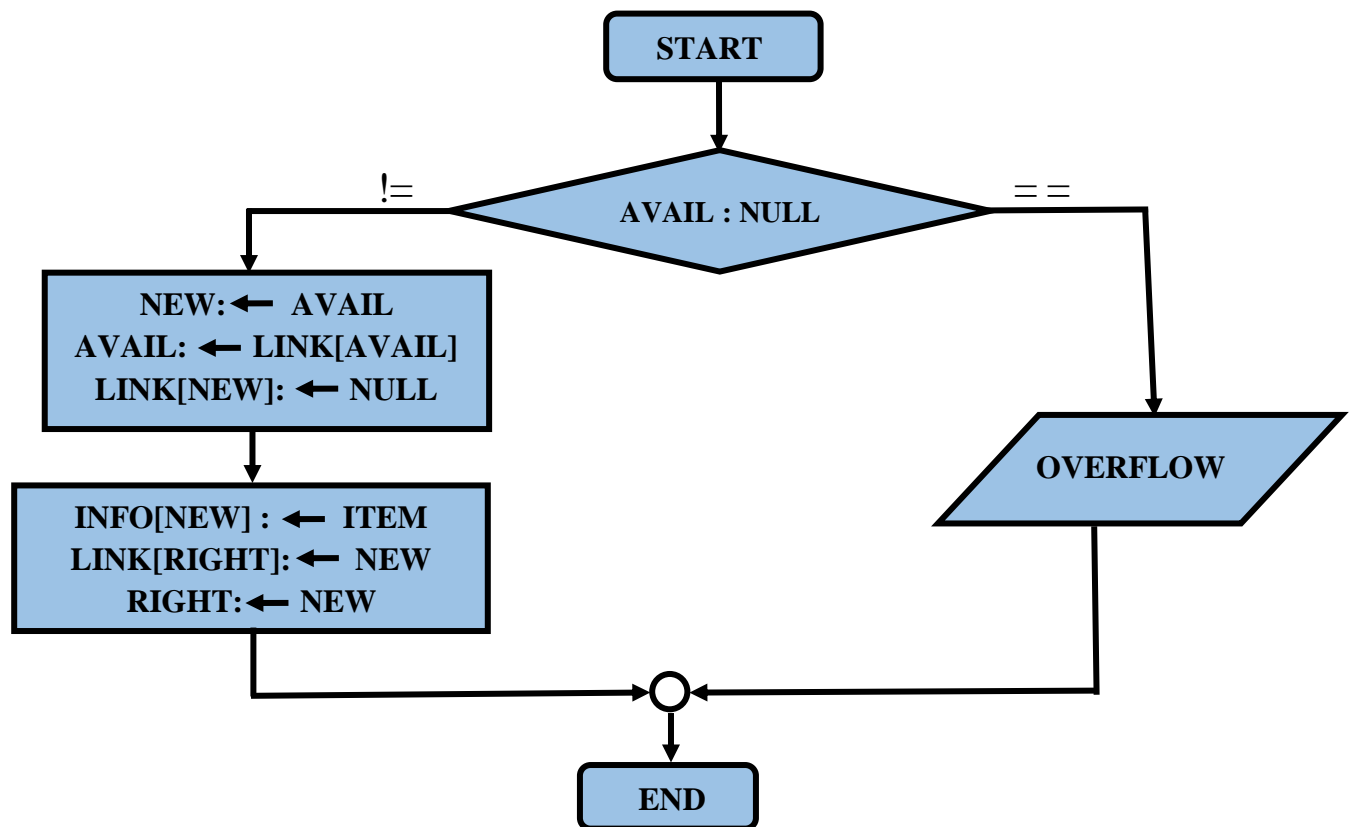
In a queue there is a pointer **REAR** that shows the **location of last data item**.

There are two operations that are normally performed on any queue :

1. **PUSH** : Inserting an element into a queue.
2. **POP** : Deleting an element from a queue.

A **Deque** is a linear list in which elements can be **added or removed** at either **end** but not in the middle. The term deque is a contraction of the name **double-ended queue**.

In a deque there is two pointer **LEFT** and **RIGHT** which points toward the last item at left side and right side.

**Problem No: 01****Problem Statement: Inserting an item into a Deque at right end.****Flow Chart :**

**Algorithm: DEQINSR(INFO, LINK, LEFT, RIGHT, AVAIL, ITEM)**

This procedure inserts an ITEM into a Deque at left end.

1. IF AVAIL:= NULL, then Write: OVERFLOW, and Return.  
[Deque already filled]
2. Else Set NEW:= AVAIL and AVAIL:= LINK[AVAIL]  
and LINK[NEW]:= NULL.
3. Set INFO[NEW]:= ITEM and LINK[RIGHT]:=NEW  
and RIGHT:= NEW.
4. Return.

**Code:**

```
#include<stdio.h>

int main(){
    int info[10]={ 10,20,30,40,50,0,0,0,0,0};
    int link[10]={ 2,3,4,5,0,7,8,9,10,0};
    int left=1,right=5,avail=6,ptr,New,item;

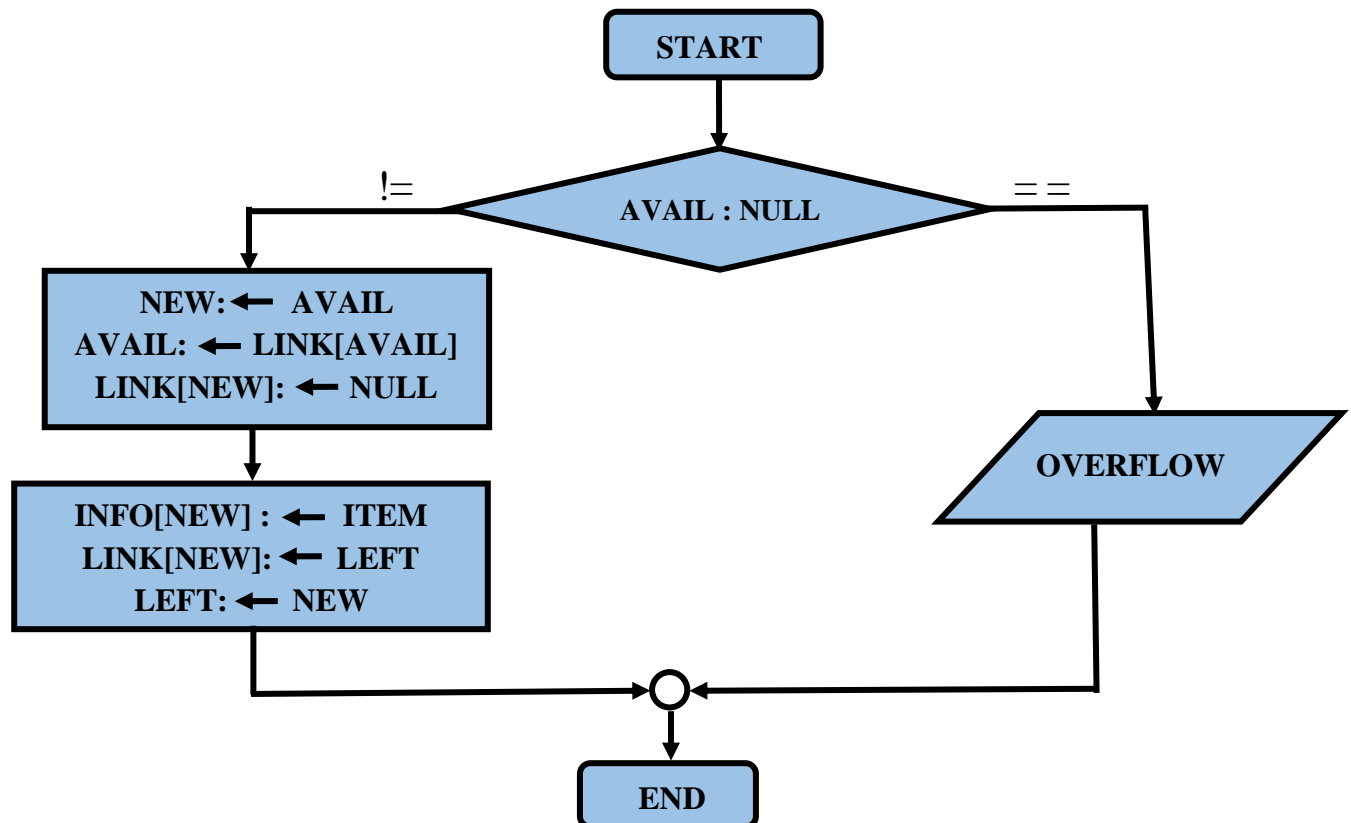
    if(avail==0)
        printf("Overflow\n");
    else{
        New=avail;
        avail=link[avail-1];
        link[New-1]=0;
        scanf("%d",&item);
        info[New-1]=item;
        link[right-1]=New;
        right=New;
    }

    return 0;
}
```

**Problem No: 02**

**Problem Statement:** Inserting an item into a Deque at left end.

**Flow Chart :**



**Algorithm: DEQINSR(INFO, LINK, LEFT, RIGHT, AVAIL, ITEM)**

This procedure insert an ITEM into a Deque at left end.

1. IF AVAIL:= NULL, then Write: OVERFLOW, and Return.  
[Deque already filled]
2. Else Set NEW:= AVAIL and AVAIL:= LINK[AVAIL]  
and LINK[NEW]:= NULL.
3. Set INFO[NEW]:= ITEM and LINK[NEW]:=LEFT  
and LEFT:= NEW.
4. Return.

**Code:**

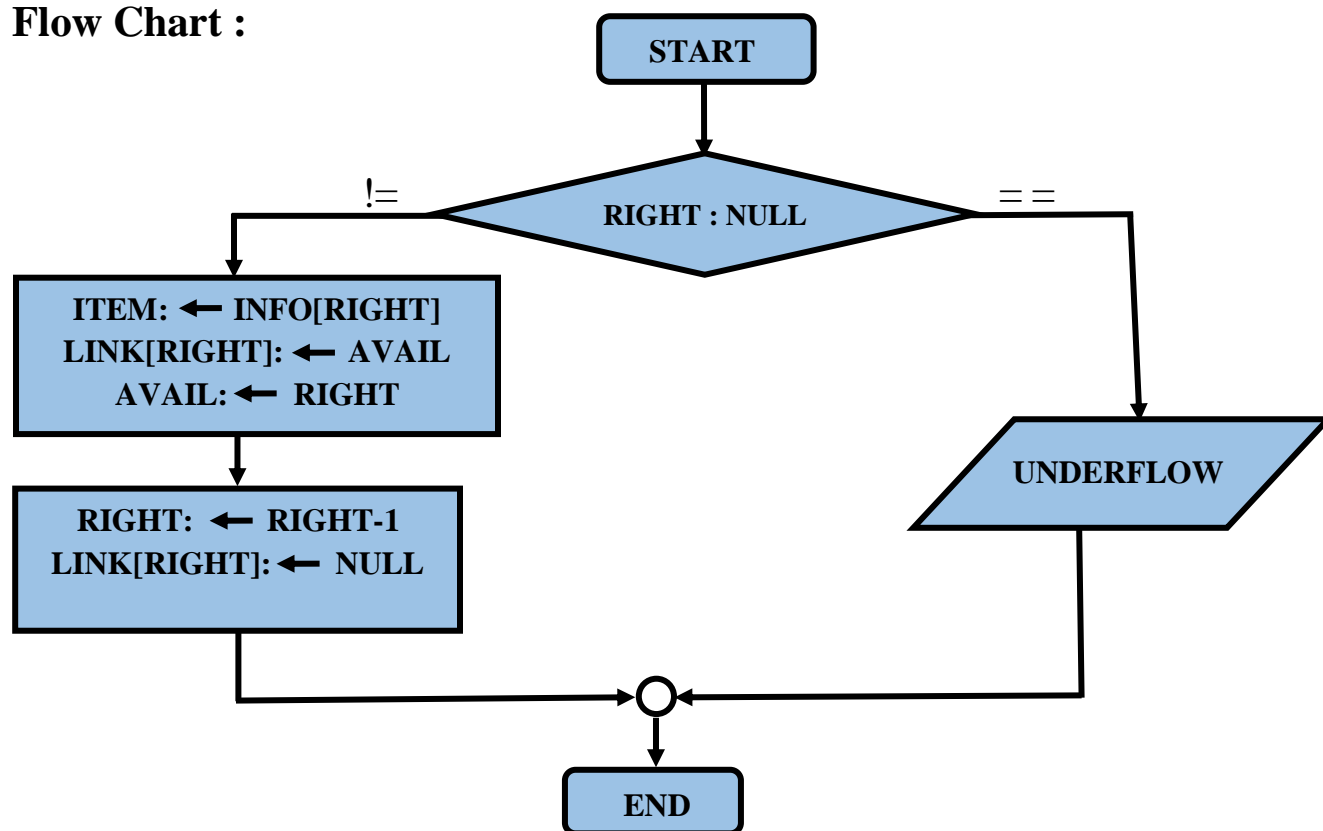
```
#include<stdio.h>

int main(){
    int info[10]={ 10,20,30,40,50,0,0,0,0,0};
    int link[10]={ 2,3,4,5,0,7,8,9,10,0};
    int left=1,right=5,avail=6,ptr,New,item;

    if(avail==0)
        printf("Overflow\n");
    else{
        New=avail;
        avail=link[avail-1];
        link[New]=0;
        scanf("%d",&item);
        info[New]=item;
        link[New]=left;
        left=New;
    }

    return 0;
```

```
}
```

**Problem No: 03****Problem Statement: Deleting an item from a Deque at right end.****Flow Chart :**



**Algorithm: DEQDEL(INFO, LINK, LEFT, RIGHT, AVAIL, ITEM)**

This procedure deletes an ITEM from a Deque at right end.

1. IF RIGHT:= NULL, then Write: UNDERFLOW, and Return.  
[Deque already empty]
2. Else Set ITEM:= INFO[RIGHT] and LINK[RIGHT]:= AVAIL  
and AVAIL:= RIGHT.
3. Set RIGHT:= RIGHT-1 and LINK[RIGHT]:= NULL.
4. Return.

**Code:**

```
#include<stdio.h>

int main(){
    int info[10]={ 10,20,30,40,50,0,0,0,0,0};
    int link[10]={ 2,3,4,5,0,7,8,9,10,0};
    int left=1,right=5,avail=6,ptr,New,item;

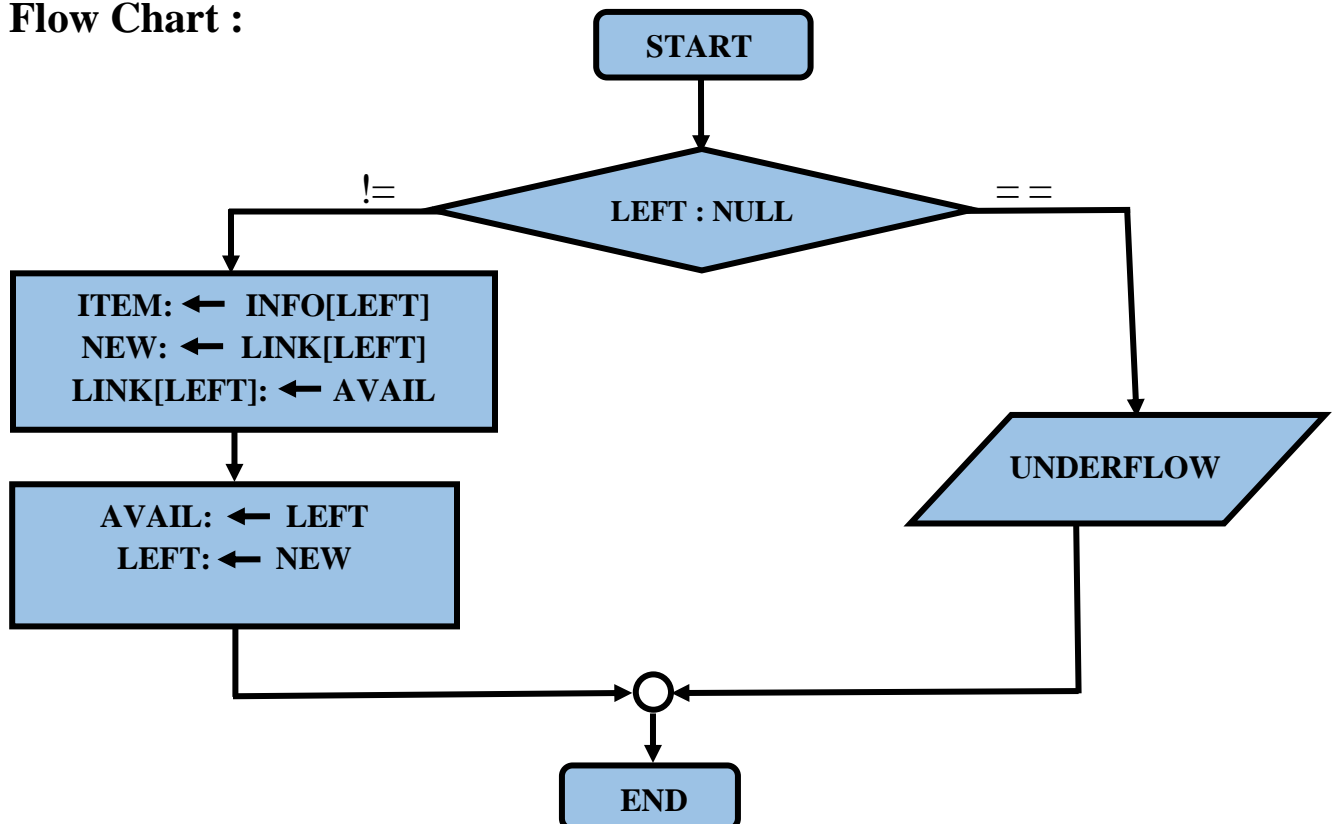
    if(right==0)
        printf("Underflow\n");
    else{
        item=info[right-1];
        link[right-1]=avail;
        avail=right;
        right=right-1;
        link[right]=0;
    }

    return 0;
}
```

**Problem No: 04**

**Problem Statement: Deleting an item from a Deque at left end.**

**Flow Chart :**



**Algorithm: DEQDELL(INFO, LINK, LEFT, RIGHT, AVAIL, ITEM)**

This procedure deletes an ITEM from a Deque at left end.

1. IF LEFT:= NULL, then Write: UNDERFLOW, and Return.  
[Deque already empty]
2. Else Set ITEM:= INFO[LEFT] and NEW:= LINK[LEFT]  
and LINK[LEFT]:= AVAIL.
3. Set AVAIL:= LEFT and LEFT:= NEW.
4. Return.

**Code:**

```
#include<stdio.h>

int main(){
    int info[10]={ 10,20,30,40,50,0,0,0,0,0};
    int link[10]={ 2,3,4,5,0,7,8,9,10,0};
    int left=1,right=5,avail=6,ptr,New,item;

    if(left==0)
        printf("Underflow\n");
    else{
        item=info[left-1];
        New=link[left-1];
        Link[left-1]=avail;
        Avail=left;
        left=New;
    }

    return 0;
}
```

**Conclusion:** I have completed all the problem with one-way link list.  
But I'm facing some problem in problem 3.

**# THE END #**