

## **Assignment No. 4**

**Name: Bhavin Ratansing Patil**

**Roll No.: 26 SEDA**

**Q.1 Implement Circular Queue using array. Perform Enqueue, Dequeue and Display operations.**

### **Queue**

Queues are more common data objects found in computer algorithms. It is a special case of more general data object, an ordered or linear list. It can be defined as an ordered list, in which all insertions are allowed at one end, called as 'rear' & all deletions are made at other end, called as 'front' i.e. First element inserted is outputted first (First In First Out or FIFO).

Queue is a linear list in which data can be inserted at one end, called the rear, and deleted from the other end, called the front. It is a first in–first out (FIFO) restricted data structure

### **Circular Queue**

In a circular array, the last element is logically followed by the first element. This is done by testing for the last element and, rather than adding one, setting the index to zero.

#### **Applications of Queues:**

1. Multiuser, multiprogramming environment job scheduling
2. Reversing stack using queue
3. queue Simulation, all types of customer service (like railway reservation) centers are designed using the concept of queues
4. Categorizing data

## Algorithms:

### Insertion in Circular queue

Step 1: if  $(\text{rear}+1) \% \text{MAX} = \text{front}$  Write "OVERFLOW " Execute step 4

Step2: [End Of if]

Step 3: if Front = -1 and Rear = -1

i. set front = rear = 0

ii. else if rear = MAX - 1 and front  $\neq$  0 set rear = 0

iii. else

1.set rear =  $(\text{rear} + 1) \% \text{MAX}$

Step 4: end if

Step 5: set QUEUE [rear] = VAL

Step 6: Exit

### Deletion in circular queue

Step 1: if (isempty\_Q())

Write " UNDERFLOW "

Goto Step 4

Step 2: End if

Step 3: set VAL = QUEUE[front]

Step 4: if front = rear

i. set front = rear = -1

Step 5: else

i. if front = MAX -1

a. set front = 0

ii. else

a. set front = front + 1

iii. end if

Step 6: end if

Step 7: exit

## Program:

### 1. Program for simple queue:

```
#include<stdio.h>
#define n 5
int main()
{
    int queue[n],ch=1,front=0,rear=0,i,j=1,x=n;
    printf("Queue using Array");
    printf("\n1.Insertion \n2.Deletion \n3.Display \n4.Exit");
    while(ch)
    {
        printf("\nEnter the Choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
                if(rear==x)
                    printf("\n Queue is Full");
                else
                {
                    printf("\n Enter no %d:",j++);
                    scanf("%d",&queue[rear++]);
                }
                break;
            case 2:
                if(front==rear)
                {
                    printf("\n Queue is empty");
                }
                else
                {
                    printf("\n Deleted Element is %d",queue[front++]);
                    x++;
                }
                break;
            case 3:
                printf("\nQueue Elements are:\n ");
                if(front==rear)
                    printf("\n Queue is Empty");
                else
                {
                    for(i=front; i<rear; i++)
                    {
                        printf("%d",queue[i]);
                        printf("\t");
                    }
                }
                break;
        }
    }
}
```

```

        case 4:
            exit(0);
        default:
            printf("Wrong Choice: please see the options");
    }
}
}
return 0;
}

```

**Output:**

```

Queue using Array
1.Insertion
2.Deletion
3.Display
4.Exit
Enter the Choice:1

    Enter no 1:11

Enter the Choice:1

    Enter no 2:22

Enter the Choice:1

    Enter no 3:33

Enter the Choice:3

Queue Elements are:
    11    22    33
Enter the Choice:2

    Deleted Element is 11
Enter the Choice:3

Queue Elements are:
    22    33
Enter the Choice:1

    Enter no 4:44

Enter the Choice:3

Queue Elements are:
    22    33    44
Enter the Choice:

```

## 2. Program for Circular queue:

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int queue[10], f = -1, r = -1;
void display();
void enqueue(int x)
{
    if ((f == 0 && r == 10 - 1) ||
        (r == (f - 1) % (10 - 1)))
    {
        printf("\nQueue is Full!");
        return;
    }
    if (f == -1)
    {
        f = 0;
        r = 0;
        queue[r] = x;
    }
    else if (r == 10 - 1 && f != 0)
    {
        r = 0;
        queue[r] = x;
    }
    else
    {
        r++;
        queue[r] = x;
    }
    display();
}
void dequeue()
{
    int x;

    if (f == -1)
    {
        printf("\nQueue is empty!");
        return;
    }
    x = queue[f];
    queue[f] = -1;
    if (f == r)
    {
        f = r = -1;
    }
    else if (f == 10 - 1)
    {

```

```

        f = 0;
    }
    else
    {
        f++;
        printf("\n%d is deleted from the queue.\n", x);
    }
    display();
}

void display()
{
    if (r == -1)
    {
        printf("\nQueue is empty");
    }
    else
    {
        printf("\nCircular Queue: ");
        if (r >= f)
        {
            for (int i = f; i <= r; i++)
                printf("%d ", queue[i]);
        }
        else
        {
            for (int i = f; i < 10; i++)
                printf("%d ", queue[i]);

            for (int i = 0; i <= r; i++)
                printf("%d ", queue[i]);
        }
    }
}

void main()
{
    int ch, x;
    do
    {
        printf("\n=====");
        printf("\n1. Enqueue\t2. Dequeue \t3. Exit ");
        printf("\nEnter your choice: ");
        scanf("%d", &ch);
        printf("\n=====");
        switch (ch)
        {
            case 1:
                printf("\nEnter Element: ");
                scanf("%d", &x);

```

```

        enqueue(x);
        break;
    case 2:
        dequeue();
        break;
    case 3:
        exit(0);
    default:
        printf("\nEnter valid choice!");
    }
} while (ch != 3);
}

```

## Output:

```

=====
1. Enqueue      2. Dequeue      3. Exit
Enter your choice: 1

=====
Enter Element: 11

Circular Queue: 11
=====
1. Enqueue      2. Dequeue      3. Exit
Enter your choice: 1

=====
Enter Element: 22

Circular Queue: 11 22
=====
1. Enqueue      2. Dequeue      3. Exit
Enter your choice: 1

=====
Enter Element: 33

Circular Queue: 11 22 33
=====
1. Enqueue      2. Dequeue      3. Exit
Enter your choice: 2

=====
11 is deleted from the queue.

Circular Queue: 22 33
=====
1. Enqueue      2. Dequeue      3. Exit
Enter your choice: 1

=====
Enter Element: 44

Circular Queue: 22 33 44
=====

```