

## **Experiment 4:** Implementation of Circular Queue Data Structure using array.

**Aim:** A circular queue is the extended version of a regular queue where the last element is connected to the first element. Thus forming a circle-like structure. Implementation of Circular Queue using Arrays is studied below.

### **Source Code:-**

```
#include <stdio.h>

#define MAX 10

int queue[MAX];
int front=-1, rear=-1;

void insert(void);
int delete_element(void);
int peek(void);
void display(void);

void insert()
{ int num;

printf("\n Enter the number to be inserted in the queue : ");
scanf("%d", &num);

if(front==0 && rear==MAX-1)
printf("\n OVERFLOW");
else if(front== -1 && rear== -1)
{ front=rear=0;
queue[rear]=num;
}

else if(rear==MAX-1 && front!=0)
```

```
{ rear=0;
queue[rear]=num;
}
else
{
rear++;
queue[rear]=num;
}}
int delete_element()
{
int val;
if(front==-1 && rear==-1)
{
printf("\n UNDERFLOW");
return -1;
}
val = queue[front];
if(front==rear)
front=rear=-1;
else
{
if(front==MAX-1)
front=0;
else
front++;
} return val;
```

```

}

int peek()
{ if(front== -1
&& rear== -1)
{
printf("\n QUEUE IS EMPTY");
return -1;
}
else
{
return queue[front];
}}

void display()
{ int i;
printf("\n");
if (front == -1 && rear == -1)
printf ("\n QUEUE IS EMPTY");
else
{
if(front<rear)
{ for(i=front;i<=rear;i++)
printf("\t%d", queue[i]);
}
else
{ for(i=front;i<MAX;i++)
printf("%d", queue[i]);
}
}
}

```

```

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

int main()
{ int option, val;
printf("D10A_Atharva Chavan_9\n");
printf("****Circular Queue Data Structure using array****\n");
printf("\n Choices are as follows:\n 1. Insert an element:\n 2. Delete an
element:\n 3. Peek:\n 4. Display the queue:\n 5. EXIT\n");
do
{
printf("\n Enter your option: ");
scanf("%d", &option);
switch(option)
{
case 1:
insert();
break;
case 2:
val = delete_element();
if(val!=-1)
printf("\n The number deleted is : %d", val);
break;
case 3:
val = peek();

```

```
if(val!=-1)
printf("\n The first value in queue is : %d", val);
break;
case 4:printf("\nElements in the Queue are:\n");
display();
break;
}
}while(option!=5);
return 0;
}
```

## Output:-

```
/tmp/HgXFG65YWU.o
D10A_Atharva Chavan_9
****Circular Queue Data Structure using array****

Choices are as follows:
1. Insert an element:
2. Delete an element:
3. Peek:
4. Display the queue:
5. EXIT

Enter your option: 1
Enter the number to be inserted in the queue : 10
Enter your option: 1
Enter the number to be inserted in the queue : 20
Enter your option: 1
Enter the number to be inserted in the queue : 30
Enter your option: 2
The number deleted is : 10
Enter your option: 3
The first value in queue is : 20
Enter your option: 4
Elements in the Queue are:

    20  30
Enter your option: 5
```