

Circular Linked List Operations

(Insertion,Deletion,Display,Search)

```
#include<stdio.h>
#include<stdlib.h>
int q[50],front=-1,rear=-1,MAX,choice,item,s;
int insert(int,int);
int deletion();
int display();
int search();
void main()
{

    printf(" \nEnter the size of the queue (LESS THAN 50)  : ");
    scanf("%d",&MAX);
    do
    {
        printf(" 1.INSERT\n 2.DELETE \n 3.DISPLAY \n 4.SEARCH \n 5.EXIT\n");
        printf(" Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 :
                printf("Input the element for insertion in queue : \n");
                scanf("%d", &item);
                insert(item,MAX);
                break;
            case 2 :
                deletion();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("enter the item to be searched for\n");
                scanf("%d",&s);
                search(s);
                break;
            case 5:
                break;
            default:
                printf("Wrong choice\n");
        }
    }while(choice!=5);
}
```

```

int insert(int item,int MAX)
{
    if((front == 0 && rear == MAX-1) || (front == rear+1))
    {
        printf("Queue Overflow \n");
        return;
    }
    if(front == -1)
    {
        front = 0;
        rear = 0;
    }
    else
    {
        if(rear == MAX-1)
            rear = 0;
        else
            rear = rear+1;
    }
    q[rear] = item ;
}

int deletion()
{
    if(front == -1)
    {
        printf("Queue Underflow\n");
        return ;
    }
    printf("Element deleted from queue is : %d\n",q[front]);
    if(front == rear)
    {
        front = -1;
        rear=-1;
    }
    else
    {
        if(front == MAX-1)
            front = 0;
        else
            front = front+1;
    }
}

int display()
{

```

```

int f = front, r = rear;
if(front == -1)
{
    printf("Queue is empty\n");
    return;
}
printf("Queue elements :\n");
if( f <= r )
while(f <= r)
{
    printf("%d ",q[f]);
    f++;
}
else
{
    while(f <= MAX-1)
    {
        printf("%d ",q[f]);
        f++;
    }
    f = 0;
    while(f <= r)
    {
        printf("%d ",q[f]);
        f++;
    }
}
printf("\n");
}

int search(int s)
{
    int i;
    for(i=front;i<=rear;i++)
    {
        if(s==q[i])
        {
            printf("element found at %d position \n",i+1);
        }
    }
}

```

OUTPUT

The image shows a Windows desktop environment. The primary application is Visual Studio Code, which is open to a C source file named 'circular_ops_array.c'. The code implements a queue using an array with operations: INSERT, DELETE, DISPLAY, SEARCH, and EXIT. The terminal window shows the execution of the program, with user input and corresponding output. The output includes prompts for queue size, element insertion, choice of operation, and the results of those operations. The Windows taskbar at the bottom contains the Start button, a search bar, and several pinned application icons. The system tray on the right shows the date and time as 07:18 PM on 06-01-2021. The Visual Studio Code status bar at the bottom indicates the current line and column (Ln 16, Col 37) and other settings like UTF-8 encoding and CRLF line endings.