

```
//Circular Queue
```

```
# include <stdio.h>
# include <conio.h>
```

```
# define SIZE 10
```

```
int front = -1, rear = -1;
```

```
int cq[SIZE];
```

```
void main()
```

```
{
    int choice;
```

```
    void insert();
    void delet();
    void display();
    void search();
```

```
    do
    {
        clrscr();
        printf("\n\t1. Insert");
        printf("\n\t2. Delete");
        printf("\n\t3. Display");
        printf("\n\t4. Search");
        printf("\n\t0. Exit");
```

```
        printf("\n\tEnter your choice : ");
        scanf("%d", &choice);
```

```
        switch(choice)
        {
            case 1:
                insert();
                break;
            case 2:
                delet();
                break;
            case 3:
                display();
                break;
            case 4:
                search();
                break;
            case 0:
                printf("\n\tEnd of Program");
                break;
            default:
                printf("\n\tInvalid Choice");
                break;
        }
        getch();
    }
```

```

while(choice != 0);
}

void insert()
{
if( (front == 0 && rear == SIZE-1) || (front==rear+1))
{
    printf("\n\tCircular Queue is full or Overflow");
}
else
{
if(rear == SIZE-1 && front > 0)
{
    rear = 0;
}
else
{
    rear++;
}

printf("\n\tEnter any number : ");
scanf("%d", &cq[rear]);

if(front == -1)
{
    front = 0;
}
}

void delet()
{
if(front == -1)
{
printf("\n\tCircular Queue is Empty or Underflow");
}
else
{
    printf("\n\tDelete Value = %d", cq[front]);

if(front == rear)
{
    front = -1;
    rear = -1;
}
else if(front == SIZE-1)
{
    front = 0;
}
else
{
    front++;
}
}
}
}

```

```

void display()
{
    int a;
    if(front == -1)
    {
        printf("\n\tCircular Queue is Empty or Underflow");
    }
    else
    {
        if(front <= rear)
        {
            for(a=front;a<=rear;a++)
            {
                printf("\n\t%d", cq[a]);
            }
        }
        else
        {
            for(a=front;a<SIZE;a++)
            {
                printf("\n\t%d", cq[a]);
            }
        }

        for(a=0;a<=rear;a++)
        {
            printf("\n\t%d", cq[a]);
        }
    }
}

```

```

void search()
{
    int a,sv, flag = 0;
    if(front == -1)
    {
        printf("\n\tCircular Queue is Empty or Underflow");
    }
    else
    {
        printf("\n\tEnter value to Search : ");
        scanf("%d", &sv);

        if(front <= rear)
        {
            for(a=front;a<=rear;a++)
            {
                if(cq[a] == sv)
                {
                    flag = 1;
                    printf("\n\tSearch value %d is found on position %d",
                        sv, a-front+1);
                    break;
                }
            }
        }
    }
}

```

```

    }
    }
else
{
    for(a=front;a<SIZE;a++)
    {
        if(cq[a] == sv)
        {
            flag = 1;
printf("\n\tSearch value %d is found on position %d",
        sv, a-front+1);
            break;
        }
    }

    if(flag == 0)
    {
        for(a=0;a<=rear;a++)
        {
            if(cq[a] == sv)
            {
                flag = 1;
printf("\n\tSearch value %d is found on position %d",
                sv, SIZE-front+a+1);
                break;
            }
        }
    }

    if(flag == 0)
    {
printf("\n\tSearch value %d does not exists", sv);
    }
}
}
}

```