

```
#include <stdio.h>

#define MAX 3

int queue[MAX];

int front = -1, rear = -1;

void insert();

int delete_element();

int peek();

void display();

int main()
{
    int option, val;

    do
    {
        printf("Enter : 1-Insert, 2-Delete, 3-Peek, 4-Display & 5-Exit : \n");

        printf("Enter your option : \n");

        scanf("%d", &option);

        switch (option)
        {
            case 1:
                insert();

                break;

            case 2:
                val = delete_element();
```

```

        if (val != -1)

            printf("The number deleted is : %d \n", val);

        break;

    case 3:

        val = peek();

        if (val != -1)

            printf("\n The first value in queue is : %d \n", val);

        break;

    case 4:

        display();

        break;

    }

} while (option != 5);

return 0;

}

```

```

void insert()

{

    int num;

    printf("Enter the number to be inserted in the queue : \n");

    scanf("%d", &num);

    if (front == 0 && rear == MAX - 1)

        printf(" OVERFLOW \n");

    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("UNDERFLOW \n");

        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("QUEUE IS EMPTY \n");

        return -1;

    }

    else

    {

        return queue[front];

    }

}

```



```

void display()
{
    int i;

    //printf("\n");

    if (front == -1 && rear == -1)

        printf("QUEUE IS EMPTY\n");

    else

    {

        if (front < rear)

        {

            for (i = front; i <= rear; i++)

                printf("%d\t", queue[i]);

        }

        else

        {

            for (i = front; i < MAX; i++)

                printf("%d\t", queue[i]);

            for (i = 0; i <= rear; i++)

                printf("%d\t ", queue[i]);

        }

        printf("\n");

    }

}

```

Enter : 1-Insert, 2-Delete, 3-Peek, 4

-Display & 5-Exit :

Enter your option :

1

Enter the number to be inserted in the queue
:

5

Enter : 1-Insert, 2-Delete, 3-Peek, 4

-Display & 5-Exit :

Enter your option :

1

Enter the number to be inserted in the queue
:

6

Enter : 1-Insert, 2-Delete, 3-Peek, 4

-Display & 5-Exit :

Enter your option :

2

The number deleted is : 5

Enter : 1-Insert, 2-Delete, 3-Peek, 4

-Display & 5-Exit :

Enter your option :

1

Enter the number to be inserted in the queue
:

8

Enter : 1-Insert, 2-Delete, 3-Peek, 4
-Display & 5-Exit :

Enter your option :

4

6 8

Enter : 1-Insert, 2-Delete, 3-Peek, 4
-Display & 5-Exit :

Enter your option :

1

Enter the number to be inserted in the queue
:

10

Enter : 1-Insert, 2-Delete, 3-Peek, 4
-Display & 5-Exit :

Enter your option :

4|

6 8 10

-