

```
#include <stdio.h>
#define MAX 10
int q[MAX], front = -1, rear = -1;
void insert();
int delete_element();
int peek();
void display();
```

```
void insert()
```

```
{
```

```
int num;
```

```
printf("In enter the no. to be inserted into queue")
```

```
scanf("%d", &num);
```

```
if (rear == MAX - 1)
```

```
    printf("Overflow");
```

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

```
{
```

```
    front = 0;
```

```
    rear = 0;
```

```
}
```

```
else
```

```
    rear++
```

```
    q[rear] = num;
```

```
}
```

```
int delete_element()
```

```
{
```

```
int val;
```

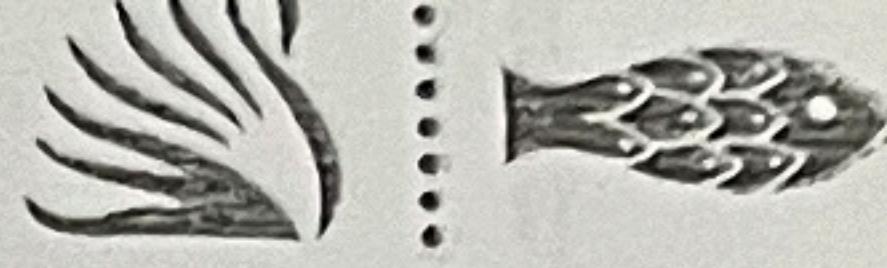
```
if (front == -1 || front > rear)
```

```
{
```

```
    printf("In underflow");
```

```
return -1 ;  
}  
else {  
    val = q[front];  
    front++;  
    if (front > rear)  
    {  
        rear = -1;  
        front = -1;  
    }  
    return val;  
}  
}  
  
int peek ()  
{  
    if (front == -1 || front > rear)  
    {  
        printf ("In queue is empty");  
        return -1;  
    }  
    else  
    return q[front];  
}  
  
void display ()  
{  
    int i;  
    printf ("In ");  
    if (front == -1 || front > rear)  
        printf ("In queue is empty");  
    else {  
        for (i = front; i <= rear; i++)
```

```
    printf("It : %d\n", q[i]);  
}  
}  
  
void main() {  
    int val, choice;  
    while (1)  
    {  
        printf("Enter your choice In 1.Insert\n2.Delete In 3.Peek\n");  
        scanf("%d", &choice);  
        switch (choice) {  
            case 1: insert();  
                break;  
            case 2: val = delete_element();  
                if (val != 1)  
                    printf("The deleted num is : %d", val);  
                break;  
            case 3: val = peek();  
                if (val != -1)  
                    printf("The first value in queue is : %d", val);  
                break;  
            case 4: display();  
                break;  
            case 5: exit(0);  
            default: printf("Default, wrong choice");  
        }  
    }  
}
```



CIRCULAR QUEUE:

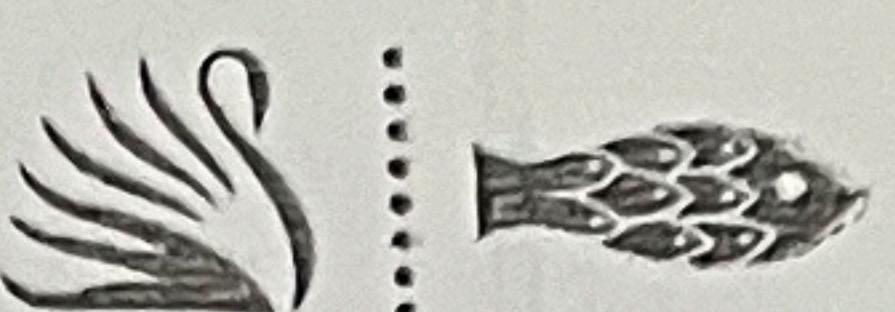
1

```
#include <stdlib.h>
#include <stdio.h>
#define SIZE 5
int items[SIZE], rear = -1, front = -1,
int isfull()
{
    if ((front == rear + 1) || (front == 0 && rear == SIZE - 1))
        return 1;
    return 0;
}
int isempty()
{
    if (front == -1)
        return 1;
    return 0;
}
void enqueue(int element)
{
    if (isfull())
    {
        printf("In queue is full");
    }
    else
    {
        if (front == -1)
            front = 0;
        rear = (rear + 1) % SIZE;
        items[rear] = element;
        printf("%d is inserted", element);
    }
}
```

```
int dequeue ()  
{  
    int value;  
    if (isEmpty ())  
    {  
        printf ("In queue is empty");  
        return -1;  
    }  
    else  
    {  
        front = (front + 1) % size  
    }  
    return (value);  
}
```

```
void display ()  
{  
    int i;  
    if (isEmpty ())  
        printf ("queue is empty");  
    else  
    {  
        printf ("In front position = %d \n", front);  
        for (i=front; i!=rear; i=(i+1)%size)  
        {  
            printf ("%d \t", items[i]);  
        }  
        printf ("%d \t", items[i]);  
    }  
}
```

```
void main ()  
{  
    int choice, element;
```



while(1)

{

printf ("1. Insert 2 Delete 3. Display 4. Exit);

printf ("Enter choice);

scanf ("%d", &choice);

switch (choice) ;

{

case 1 : printf ("In Enter the element to insert ");

scanf ("%d", &element);

enqueue (element);

break;

case 2 : element = deQueue ();

if (element != -1)

printf ("Element is deleted ", element);

break;

case 3 : ~~display ()~~

break;

case 4 : exit (0);

default : printf ("In Invalid choice ");

}

}

OUTPUT :

1. The operation to be performed

1. Insert

2. Delete

3. Display

choice : 1

5 deleted