```
#include <stdio.h>

#define n 3
int que[n] ,front=-1 ,rear=-1 ;


void enqueue(int que[] ,int* front ,int* rear ){
        if( *rear ==n-1)
        {
                printf("Queue is full\n");
                return;
        }
        if( *front ==-1 )
        {
                *front=*front+1;
        }
        int item;
        printf("What should I insert? :  ");
    scanf("%d",&item);

        *rear=*rear+1;
        que[*rear]=item;

        //printf("**%d,%d**",*front,*rear);
}

void dequeue(int que[] ,int* front ,int* rear ){
        int item;
        if( *front==-1 && *rear==-1 ){
            printf("Empty Queue\n");
        }
        else{
        item=que[*front];
                if( *front==*rear ){
                        *rear=-1;
                        *front=-1;
                }
                else{
                        *front=*front+1;
                //que[*front]='\0';
                }
        printf("%d is removed\n",item);
        }
        //printf("**%d,%d**",*front,*rear);
}

void display(int que[] ,int front ,int rear){
        if( front==-1 && rear==-1 ){
                printf("Queue is empty\n");
        }
```

```
        else
        {
                for (int i=front ;i<=rear ;i++){
                        printf("%d\t",que[i]);
                }
                printf("\n");
        }
        //printf("**%d,%d**",front,rear);
}

int main(){
        //printf("Enter the total size of the Queue : ");
    //int n=3;
    //scanf("%d", & n);
    int choice;
    int item=0;
    printf("1...display\n");
    printf("2...enque\n");
    printf("3...deque\n");
    printf("4...quit\n");

    int quit=1;
    while(quit!=0){
        printf("\nOption : ");
        scanf("%d",&choice);
        switch(choice){
                case 1: display(que,front,rear);
                        break;
                case 2: enqueue(que,&front,&rear);
                        break;
                case 3: dequeue(que,&front,&rear);
                        break;
                case 4: quit=0;
        }
    }

    return 0;
}
```

```
> ./main
1...display
2...enque
3...deque
4...quit

Option : 2
What should I insert? :  1

Option : 2
What should I insert? :  2

Option : 2
What should I insert? :  3

Option : 1
1   2   3

Option : 2
Queue is full

Option : 3
1 is removed

Option : 3
2 is removed

Option : 1
3

Option : 3
3 is removed

Option : 3
Empty Queue
```

7 _ Queue

Aim

To impliment queue data structure.

Algorithm

ENQUEUE(ITEM)

1. START
2. If (REAR == N-1) then
    1. print "Queue Full"
3. Else
    1. If (FRONT == -1) then
        1. FRONT = FRONT+1
        2. REAR = REAR+1
    2. Q[REAR] = ITEM
4. end If
5. STOP

DEQUEUE ( )

1. START
2. If (FRONT == REAR-1) Then
    1. Print "Queue Empty"
3. Else
    1. Item = Q[FRONT]
    2. If (FRONT == REAR)
        1. REAR = -1
        2. FRONT = -1
    3. Else
        1. FRONT = FRONT + 1
    4. End if.
4. End If
5. STOP