DSA SERIES

- Learn Coding

Topic to be Covered today

Queue Problems

LETS START TODAY'S LECTURE

C++ STL Queue (GFG)

```
void push(queue<int> &q, int x) {
 q.push(x);

}

int pop(queue<int> &q) {

   if(!q.empty()){
       int front  = q.front();
       q.pop();
       return front;
   }
   return -1;
}

int getSize(queue<int> &q) {
   return q.size();
}
```

```
int getBack(queue<int> &q) {
    if(!q.empty()){
        return q.back();
    }
}

int getFront(queue<int> &q) {

    if(!q.empty()){
        return q.front();
    }
    return -1;
}
```

Implement Stack Using Queue(225)

```
class MyStack {
public:
    queue<int> q;
    MyStack() {}

    void push(int x) {
        int n = q.size();
        q.push(x);

        for (int i = 0; i < n; i++) {
            q.push(q.front());
            q.pop();
        }
    }

    int pop() {
        if (q.empty()) {
            return -1;
        }
```

```
        int n = q.front();
        q.pop();
        return n;
    }

    int top() {
        if (q.empty()) {
            return -1;
        }
        return q.front();
    }

    bool empty() {
        if (q.size() == 0) {
            return true;
        }
        return false;
    }
};
```

Implement Queue Using Stacks(225)

```
class MyQueue {
public:
    stack<int> s1;
    stack<int> s2;
    int count;
    MyQueue() { count = 0; }

    void push(int x) {
        s1.push(x);
        count++;
    }

    int pop() {
        if (s2.empty()) {
            while (!s1.empty()) {
                int d = s1.top();
                s1.pop();
                s2.push(d);
            }
```

```
}
    count--;
    int x = s2.top();
    s2.pop();
    return x;
}

int peek() {
    if (s2.empty()) {
        while (!s1.empty()) {
            int d = s1.top();
            s1.pop();
            s2.push(d);
        }
    }

    int x = s2.top();
    return x;
```

```
    }

    bool empty() {
        if (count == 0) {
            return true;
        } else {
            return false;
        }
    }
};
```

Implement Queue Using Linked List(GFG)

```
void MyQueue:: push(int x)
{
        QueueNode *newNode = new QueueNode(x);
        if(rear == NULL){
            front = rear = newNode;
        } else {
            rear -> next = newNode;
            rear = newNode;
        }
}

//Function to pop front element from the queue.
int MyQueue :: pop()
{
    if(front == NULL){
        return -1;
    }
```

```
 int val = front->data;
    QueueNode* temp = front;
    front = front ->next;

    if(front == NULL){
       rear = NULL;
    }
    delete temp ;
    return val;
}
```

Design Circular Queue(622)

```
class MyCircularQueue {
public:
int *arr;
int front ,rear ,size ,capacity;
    MyCircularQueue(int k) {
        capacity = k;
        arr = new int[capacity];
        front = 0;
        rear =0;
        size=0;

    }

    bool enQueue(int value) {
        if(size == capacity){
            return false;
        }
        arr[rear]=value;
        rear = (rear+1)%capacity;
        size++;
        return true;
```

```
}

bool deQueue() {
    if(size == 0){
        return false;
    }

    front = (front+1)%capacity;
    size--;
    return true;

}

int Front() {

    if(size == 0){
        return -1;
    }
    return arr[front];

}
```

```
  int Rear() {
        if(size==0){
            return -1;
        }

        return arr[(rear-1+capacity)%capacity];

    }

    bool isEmpty() {
        return size==0;

    }

    bool isFull() {
        return size==capacity;

    }
};
```

Reverse first k of a Queue(GFG)

```
class Solution {
  public:
    queue<int> reverseFirstK(queue<int> q, int k) {

        // Step 1 :
    if(q.size()<k) return q;


    //   step 2 :
       stack<int> st;
       for(int i =0 ;i<k;i++){
           st.push(q.front());
           q.pop();
       }

       // Step 3:
       while(!st.empty()){
           q.push(st.top());
           st.pop();
       }
```

```
 // step 4:

        int rem = q.size()-k;
        for(int i =0;i<rem;i++){
            q.push(q.front());
            q.pop();
        }

        return q;


    }
};
```

Learn coding

THANK YOU