**Day-21**

**Problem1**-   
[**232. Implement Queue using Stacks**](https://leetcode.com/problems/implement-queue-using-stacks/)

class MyQueue {

public:

stack<int> st1, st2;

    void push(int x) {

        st1.push(x);

    }

    int pop() {

        while(!st1.empty())

        {

            st2.push(st1.top());

            st1.pop();

        }

        int ans = st2.top();

        st2.pop();

        while(!st2.empty())

        {

            st1.push(st2.top());

            st2.pop();

        }

        return ans;

    }

    int peek() {

        while(!st1.empty())

        {

            st2.push(st1.top());

            st1.pop();

        }

        int ans = st2.top();

        while(!st2.empty())

        {

            st1.push(st2.top());

            st2.pop();

        }

        return ans;

    }

    bool empty() {

        return st1.empty();

    }

};

**Problem2**- [**496. Next Greater Element I**](https://leetcode.com/problems/next-greater-element-i/)

class Solution {

public:

vector<int> nextGreaterElement(vector<int>& nums1, vector<int>& nums2) {

int n=nums1.size();

int mm=nums2.size();

unordered\_map<int,int>m;

for(int i=0;i<n;i++)

{

m[nums1[i]]=i;

}

stack<int>s;

int i=0;

vector<int>ans(n,-1);

s.push(nums2[i]);

for(int i=1;i<mm;i++)

{

while(!s.empty()&& nums2[i]>s.top())

{

int x=s.top();

s.pop();

if(m.find(x)!=m.end())

{

ans[m[x]]=nums2[i];

}

}

s.push(nums2[i]);

}

return ans;

}

};**Problem 3**- [**48. Rotate Image**](https://leetcode.com/problems/rotate-image/)

class Solution {

public:

void rotate(vector<vector<int>>& matrix) {

int n=matrix.size();

for(int i=0;i<n-1;i++)

{

for(int j=i+1;j<n;j++)

{

swap(matrix[i][j],matrix[j][i]);

}

}

for(int i=0;i<n;i++)

{

int p1=0,p2=n-1;

while(p1<p2)

{

swap(matrix[i][p1],matrix[i][p2]);

p1++;

p2--;

}

}

}

};