Implementation of a stack using array

Stack is based on last in first out scheme.

We maintain a global array and a top element for the array which is used to perform all the oper

Initially : top=-1;

Push :top++; st[top]=elem;

pop : top- -; return st[top+1];

isEmpty(): if(top==-1) return 1; else return 0;

constructor : just initialise default values.

```
#include <bits/stdc++.h>
// Stack class.
class Stack {

public:
    vector<int> st;
    int tp;
    int n;
    Stack(int capacity) {
        // Write your code here.
        this->st.resize(capacity);
        this->tp=-1;
        this->n=capacity;
    }

    void push(int num) {
        // Write your code here.
        if(tp!=n-1)
        {
            tp++;
            st[tp]=num;
        }

    }

    int pop() {
        // Write your code here.
        if(tp==-1)
        {
            return -1;
        }
        tp--;
        return st[tp+1];
    }

    int top() {
        // Write your code here.
        if(tp!=-1)
            return st[tp];
        return -1;
```

```
    }

    int isEmpty() {
        // Write your code here.
        if(tp==-1)
            return 1;
        else
            return 0;

    }

    int isFull() {
        // Write your code here.
        if(tp==n-1)
            return 1;
        else
            return 0;
    }

};
```