Additional Questions:

1]

#include<iostream>

#include<stack>

#include<vector>

using namespace std;

vector<int> nearestSmaller(vector<int> &arr){

stack<int> st;

vector<int> result;

for(int i=0;i<arr.size();i++){

while (!st.empty() && st.top() >= arr[i]){

st.pop();

}

if(st.empty()) result.push\_back(-1);

else result.push\_back(st.top());

st.push(arr[i]);

}

return result;

}

int main(){

vector<int> arr={4, 5, 2, 10, 8};

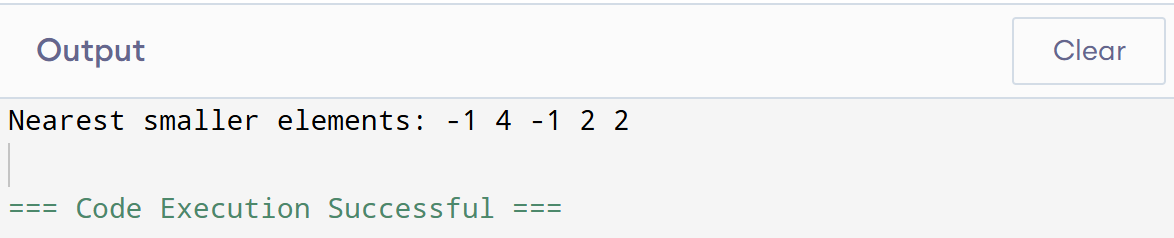
vector<int> ans=nearestSmaller(arr);

cout << "Nearest smaller elements: ";

for(int x:ans) cout << x << " ";

return 0;

}



2]

#include<iostream>

#include<stack>

using namespace std;

class SpecialStack{

stack<int> st;

int minElement;

public:

void push(int x){

if(st.empty()){

st.push(x);

minElement=x;

}else{

if(x >= minElement){

st.push(x);

}else{

st.push(2\*x - minElement);

minElement=x;

}

}

}

void pop(){

if(st.empty()) return;

int t=st.top();

st.pop();

if(t<minElement){

minElement=2\*minElement-t;

}

}

int top(){

int t=st.top();

if(t<minElement) return minElement;

return t;

}

int getMin(){

return minElement;

}

};

int main(){

SpecialStack s;

s.push(3);

s.push(5);

cout << "Min: " << s.getMin() << endl;

s.push(2);

s.push(1);

cout << "Min: " << s.getMin() << endl;

s.pop();

cout << "Min: " << s.getMin() << endl;

s.pop();

cout << "Top: " << s.top() << endl;

cout << "Min: " << s.getMin() << endl;

}



3]

#include<iostream>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

if(n%2==0) cout << "Even\n";

else cout << "Odd\n";

}

return 0;

}

