**#include <bits/stdc++.h>**

**using namespace std;**

**// Stack Classical Problems**

**void populateNextSmaller(const int arr[], vector<int>&nextSmaller) {**

**int n = nextSmaller.size();**

**stack<int>stk;**

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

**if(stk.empty() or arr[stk.top()] <= arr[i]) {**

**stk.push(i);**

**} else {**

**while(!stk.empty() and arr[stk.top()] > arr[i]) {**

**nextSmaller[stk.top()] = i;**

**stk.pop();**

**}**

**stk.push(i);**

**}**

**}**

**while(!stk.empty()) {**

**nextSmaller[stk.top()] = n;**

**stk.pop();**

**}**

**}**

**void populatePrevSmaller(const int\* arr, vector<int>&prevSmaller) {**

**int n = prevSmaller.size();**

**stack<int>stk;**

**for(int i=n-1;i>=0;i--) {**

**if(stk.empty() or arr[stk.top()] <= arr[i]) {**

**stk.push(i);**

**} else {**

**while(!stk.empty() and arr[stk.top()] > arr[i]) {**

**prevSmaller[stk.top()] = i;**

**stk.pop();**

**}**

**stk.push(i);**

**}**

**}**

**while(!stk.empty()) {**

**prevSmaller[stk.top()] = -1;**

**stk.pop();**

**}**

**}**

**void insertAtBottom(int element, stack<int>&stk) {**

**if(!stk.empty() ) {**

**int top = stk.top();**

**stk.pop();**

**insertAtBottom(element, stk);**

**stk.push(top);**

**} else {**

**stk.push(element);**

**}**

**}**

**void reverseStack(stack<int>&stk) {**

**if(!stk.empty()) {**

**int top = stk.top();**

**stk.pop();**

**reverseStack(stk);**

**insertAtBottom(top, stk);**

**}**

**}**

**int main() {**

**// Next Greater Element**

**int arr[] = {7, 12, 4 ,6, 12, 13, 13};**

**int ans[7];**

**stack<int>stk;**

**int n = 7;**

**for(int i=0;i< n;i++) { // Progress in the array**

**if(stk.empty() or arr[stk.top()] >= arr[i]) {**

**stk.push(i);**

**} else {**

**while(!stk.empty() and arr[stk.top()] < arr[i]) {**

**ans[stk.top()] = arr[i];**

**stk.pop();**

**}**

**stk.push(i);**

**}**

**}**

**while(!stk.empty()) {**

**ans[stk.top()] = -1;**

**stk.pop();**

**}**

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

**cout<<arr[i]<<" ans is: "<< ans[i]<<endl;**

**}**

**int histograms[] = {2, 7, 12, 13, 6, 9, 10};**

**/\***

**\* 2 -> 14**

**\* 7-> 21**

**\* 12 -> 24**

**\* 13 -> 13**

**\* 6-> 36 // ans**

**\* 9-> 18**

**\* 10->10**

**\*/**

**vector<int>next(7), prev(7);**

**n = 7;**

**populateNextSmaller(histograms, next);**

**populatePrevSmaller(histograms, prev);**

**int mxAreaRectangle = -1;**

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

**mxAreaRectangle = max(mxAreaRectangle, (next[i] - prev[i] - 1) \* histograms[i]);**

**}**

**cout<<"The maximum histogram area is: "<<mxAreaRectangle<<endl;**

**// Reverse a stack**

**stack<int>rStk;**

**for(int i=1;i<=5;i++){**

**rStk.push(i);**

**}**

**reverseStack(rStk);**

**cout<<"After reversing stack"<<endl;**

**while(!rStk.empty()) {**

**cout<<rStk.top()<<" ";**

**rStk.pop();**

**}**

**cout<<endl;**

**return 0;**

**}**