**link=**[**https://www.youtube.com/watch?v=FRJrfCjC2GY&list=PLoa\_roVVsxA1CJu4DsOljb9d7FJkMCynM&index=11**](https://www.youtube.com/watch?v=FRJrfCjC2GY&list=PLoa_roVVsxA1CJu4DsOljb9d7FJkMCynM&index=11)

**[L1]Class - 11 ( STL Stack and practice problems ) [Bangla]**

Class Video link :

[https://youtu.be/FRJrfCjC2GY](https://youtu.be/FRJrfCjC2GY?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

Discussed topics :

1) STL Stack & Practice problems

2) balanced brackets ( Hacker Rank )

Link : [https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/balanced-brackets-3-4fc590c6/](https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/balanced-brackets-3-4fc590c6/?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

My Code : [https://paste.ubuntu.com/p/B2KSr2QqGt/](https://paste.ubuntu.com/p/B2KSr2QqGt/?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

3) Special Stack ( Geeks for Geeks )

Link : [https://practice.geeksforgeeks.org/problems/special-stack/1](https://l.facebook.com/l.php?u=https%3A%2F%2Fpractice.geeksforgeeks.org%2Fproblems%2Fspecial-stack%2F1%3Ffbclid%3DIwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw&h=AT3hp8lcfbkmRZtbIyf3oBC2J2hrC3wE0SK5YcKh8OdC4rMyjayLkJg4GlRizg0Tqr7lJ49ur9rn3hAmC-5ztDG7d7mzZy2yt1yEWAJoSWTI9QQDMO05eiRYrEdw&__tn__=-UK-R&c%5B0%5D=AT1JRFZTaRxn4qBaXox2oCQOy5uHbNvDaAfVhgrqbxGDcO18TxZAjnCDhfTCVDssdMEQ2Tl1Kz1ihiCzGkuNTJy3bce7VyX653YdqehdTeh-Fc29MVpiw0_LmM_ABcRqxA_AeJzANH0kKR0AnDANhvq4ehs9FLnIvrG2mC6X41QYNd_sc2up_npkXw)

My Code :[https://paste.ubuntu.com/p/yYHBWx5gRb/](https://paste.ubuntu.com/p/yYHBWx5gRb/?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

4) 682. Baseball Game ( LeetCode)

Link : [https://leetcode.com/problems/baseball-game/](https://leetcode.com/problems/baseball-game/?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

My Code :[https://paste.ubuntu.com/p/ryzgZs75fF/](https://paste.ubuntu.com/p/ryzgZs75fF/?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

5) 1047. Remove All Adjacent Duplicates In String ( LeetCode )

Link : [https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/](https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

My Code : [https://paste.ubuntu.com/p/BXJttd6ZYn/](https://paste.ubuntu.com/p/BXJttd6ZYn/?fbclid=IwAR3_AVk3GIc7_ThVoeZVeIHHlPuFIrb_HTpaFD_WHsJbrQy4rKmhmr9BfFw)

Class Tasks :

All the problems discussed in class are class tasks.

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

**using namespace std;**

***#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);***

***#define endl '\n'***

**int main()**

**{**

**optimize();**

**stack<int>st;**

**st.push(1);**

**st.push(2);**

**st.push(3);**

**st.push(4);**

**st.push(5);**

**cout<<"SIZE:"<<st.size()<<endl;**

**cout<<st.top()<<endl;//5**

**st.pop();//top element delete**

**cout<<st.top()<<endl;//4**

**cout<<"Stack all element print:"<<endl;**

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

**cout<<st.top()<<endl;**

**st.pop();**

**}//4 3 2 1**

**return 0;**

**}**

**"C:\Users\Md.Hamid Hosen\CLionProjects\untitled\cmake-build-debug\untitled.exe"**

**SIZE:5**

**5**

**4**

**Stack all element print:**

**4**

**3**

**2**

**1**

**Process finished with exit code 0**

**problem link=**[**https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/balanced-brackets-3-4fc590c6/description/?sort=recent-comments**](https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/balanced-brackets-3-4fc590c6/description/?sort=recent-comments)

**code link=**[**https://paste.ubuntu.com/p/B2KSr2QqGt/**](https://paste.ubuntu.com/p/B2KSr2QqGt/)

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

**using namespace std;**

***#define endl '\n'***

***#define optimeze() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);***

**bool isBalance(char c1,char c2){**

**return ((c1=='{' && c2=='}')||(c1=='[' && c2==']')||(c1=='(')&&c2==')');**

**}**

**int main()**

**{**

**optimeze();**

**int t;**

**cin>>t;**

**while(t--){**

**string s;**

**cin>>s;**

**stack<char>st;**

**bool done=1;**

**for(auto u:s){**

**if(u=='{'||u=='['||u=='('){**

**st.push(u);**

**}else{**

**if(st.empty()){**

**done=0;**

**break;**

**}else{**

**if(isBalance(st.top(),u)){**

**st.pop();**

**}else{**

**done=0;**

**break;**

**}**

**}**

**}**

**}**

**if(!st.empty()) done=0;**

**if(done) cout<<"YES"<<endl;**

**else cout<<"NO"<<endl;**

**}**

**return 0;**

**}**

**"C:\Users\Md.Hamid Hosen\CLionProjects\untitled\cmake-build-debug\untitled.exe"**

**3**

**{[()]}**

**{[(])}**

**{{[[(())]]}}**

**YES**

**NO**

**YES**

**Process finished with exit code 0**

**problem link=**[**https://practice.geeksforgeeks.org/problems/special-stack/1**](https://practice.geeksforgeeks.org/problems/special-stack/1)

**code link=**[**https://paste.ubuntu.com/p/yYHBWx5gRb/**](https://paste.ubuntu.com/p/yYHBWx5gRb/)

**// { Driver Code Starts**

***#include<iostream>***

***#include<stack>***

**using namespace std;**

**void push(int a);**

**bool isFull(int n);**

**bool isEmpty();**

**int pop();**

**int getMin();**

**//This is the STL stack (http://quiz.geeksforgeeks.org/stack-container-adaptors-the-c-standard-template-library-stl/).**

**stack<int> s;**

**int main(){**

**int t;**

**cin>>t;**

**while(t--){**

**int n,a;**

**cin>>n;**

**while(!isEmpty()){**

**pop();**

**}**

**while(!isFull(n)){**

**cin>>a;**

**push(a);**

**}**

**cout<<getMin()<<endl;**

**}**

**}// } Driver Code Ends**

**/\*Complete the function(s) below\*/**

**void push(int a)**

**{**

**//add code here.**

**s.push(a);**

**}**

**bool isFull(int n)**

**{**

**//add code here.**

**return n==s.size();**

**}**

**bool isEmpty()**

**{**

**//add code here.**

**return s.empty();**

**}**

**int pop()**

**{**

**//add code here.**

**s.pop();**

**}**

**int getMin()**

**{**

**//add code here.**

**int x=2e9;**

**//int x=100000;**

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

**x=min(s.top(),x);**

**s.pop();**

**}**

**return x;**

**}**

**problem link=**[**https://leetcode.com/problems/baseball-game/submissions/**](https://leetcode.com/problems/baseball-game/submissions/)

**code link=**[**https://paste.ubuntu.com/p/ryzgZs75fF/**](https://paste.ubuntu.com/p/ryzgZs75fF/)

**class Solution {**

**public:**

**int calPoints(vector<string>& ops) {**

**stack<int> st;**

**for ( auto u : ops ) {**

**if ( u.size() > 1 ) {**

**st.push ( stoi ( u ) );**

**}**

**else if ( isdigit ( u[0] ) ) {**

**st.push ( u[0] - '0' );**

**}**

**else if ( u[0] == '+' ) {**

**int a1 = -1, a2 = -1;**

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

**a1 = st.top();**

**st.pop();**

**}**

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

**a2 = st.top();**

**st.pop();**

**}**

**if ( a2 != -1 ) {**

**st.push ( a2 );**

**st.push ( a1 );**

**st.push ( a1+a2 );**

**}**

**else if ( a1 != -1 ) {**

**st.push ( a1 );**

**}**

**}**

**else if ( u[0] == 'D' ) {**

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

**st.push ( 2 \* st.top() );**

**}**

**}**

**else {**

**if ( !st.empty() ) st.pop();**

**}**

**}**

**long long ans = 0;**

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

**ans += st.top();**

**st.pop();**

**}**

**return ans;**

**}**

**};**

**["5","2","C","D","+"]**

**30**

**problem link=**[**https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/submissions/**](https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/submissions/)

**code link=**[**https://paste.ubuntu.com/p/BXJttd6ZYn/**](https://paste.ubuntu.com/p/BXJttd6ZYn/)

**class Solution {**

**public:**

**string removeDuplicates(string s) {**

**stack <char> st;**

**for ( auto u : s ) {**

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

**if ( !st.empty() && st.top() == u ) st.pop();**

**else st.push(u);**

**}**

**else st.push(u);**

**}**

**s.clear();**

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

**s += st.top();**

**st.pop();**

**}**

**reverse ( s.begin(), s.end() );**

**return s;**

**}**

**};**

**abbaca**

**ca**