Problem | Editorial | Submissions | Comments

Implement stack using array

Basic Accuracy: 54.76% Submissions: 117K+ Points: 1

Write a program to implement a Stack using Array. Your task is to use the class as shown in the comments in the code editor and complete the functions push() and pop() to implement a stack.

Example 1:

Input:
push(2)
push(3)
pop()
push(4)
pop()
Output: 3, 4
Explanation:
push(2) the stack will be {2}
push(3) the stack will be {2 3}

C++ (g++ 5.4) Average Time: 25m Start Timer

```
// } Driver Code Ends



void MyStack :: push(int x)
{
    // Your Code
    top++;
    arr[top] = x;
}

//Function to remove an item from top of the stack.
int MyStack :: pop()
{
    // Your Code
    if(top == -1) return -1;
    else{
        top--;
        return arr[top+1];
    }
}
```

Custom Input Compile & Run Submit

Problem | Editorial | Submissions | Comments

Implement Stack using Linked List

Basic Accuracy: 53.98% Submissions: 96K+ Points: 1

Let's give it a try! You have a linked list and you have to implement the functionalities push and pop of stack using this given linked list. Your task is to use the class as shown in the comments in the code editor and complete the functions push() and pop() to implement a stack.

Example 1:

Input:
push(2)
push(3)
pop()
push(4)
pop()
Output: 3 4
Explanation:
push(2) the stack will be {2}

C++ (g++ 5.4) Average Time: 40m Start Timer

```
// } Driver Code Ends


//Function to push an integer into the stack.
void MyStack ::push(int x)
{
    StackNode *temp = new StackNode(x);
    temp ->next = top;
    top = temp;
}

//Function to remove an item from top of the stack.
int MyStack ::pop()
{
    if(top == NULL) return -1;

    int ans = top->data;
    StackNode *temp = top;
    top = top->next;
    delete(temp);
    return ans;

}
```

Custom Input | Compile & Run | Submit

Problem | Editorial | Submissions | Comments

Parenthesis Checker

Easy Accuracy: 28.56% Submissions: 433K+ Points: 2

Given an expression string **x**. Examine whether the pairs and the orders of {,},(,),[,] are correct in exp.
For example, the function should return 'true' for exp = [()]{}{[()()]()} and 'false' for exp = [(]).

Note: The drive code prints "balanced" if function return true, otherwise it prints "not balanced".

Example 1:

Input:
{([])}
Output:
true
Explanation:
{ ([ ]) }. Same colored brackets can form

C++ (g++ 5.4) | Start Timer

```
// } Driver Code Ends

class Solution
{
    public:
    //Function to check if brackets are balanced or not.
    bool ispar(string x)
    {
        // Your code here
        int n = x.size();
        stack<int>s;
        for(int i = 0 ; i < n ; i++){
            if(s.empty()){
                s.push(x[i]);
            }else if(x[i] == ']' && s.top() == '[' ||
                     x[i] == ')' && s.top() == '(' ||
                    x[i] == '}' && s.top() == '{' ){
                s.pop();
            }else{
                s.push(x[i]);
            }
        }
        if(s.empty()) return 1;
        else return 0;
    }

};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Problem | Editorial | Submissions | Comments

Get minimum element from stack

Medium Accuracy: 22.59% Submissions: 207K+ Points: 4

You are given **N** elements and your task is to Implement a Stack in which you can get minimum element in O(1) time.

Example 1:

Input:
push(2)
push(3)
pop()
getMin()
push(1)
getMin()
Output: 2 1
Explanation: In the first test case for query
push(2) Insert 2 into the stack

C++ (g++ 5.4) Average Time: 30m Start Timer

```
class Solution{
    int minEle;
    stack<int> s , a ;

    public:

       /*returns min element from stack*/
       int getMin(){
          if(s.empty()) return -1;

          int mini = a.top();
          return mini;
       }

       /*returns poped element from stack*/
       int pop(){
           if(s.empty()) return -1;
           if(a.top() == s.top()){
               a.pop();
           }
           int p = s.top();
           s.pop();
           return p;
       }

       /*push element x into the stack*/
       void push(int x){
          s.push(x);
          if(a.size() == 0 || a.top() >= s.top()){
              a.push(x);
          }
       }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit