Actividad 2.1

Programación de Estructuras de Datos y Algoritmos Fundamentales

Alexa Serrano Negrete - A01654063
Luis Manuel Orozco Yáñez - A01707822
Leonardo Millán Velázquez - A01639823

Success Details ›

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Reverse Integer.

Memory Usage: 5.9 MB, less than 34.10% of C++ online submissions for Reverse Integer.

Next challenges:

String to Integer (atoi) | Reverse Bits

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
09/20/2021 18:57	Accepted	0 ms	5.9 MB	cpp

```cpp
class Solution{
public:
    int reverse(int x) {
        int rev = 0;
        while (x != 0) {
            int pop = x % 10;
            x /= 10;
            if (rev > INT_MAX/10 || (rev == INT_MAX / 10 && pop > 7)) return 0;
            if (rev < INT_MIN/10 || (rev == INT_MIN / 10 && pop < -8)) return 0;
            rev = rev * 10 + pop;
        }
        return rev;
    }
};
```

Success Details ›

Runtime: 30 ms, faster than 10.34% of C++ online submissions for Palindrome Number.

Memory Usage: 5.9 MB, less than 30.85% of C++ online submissions for Palindrome Number.

Next challenges:

Palindrome Linked List

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
09/20/2021 18:56	Accepted	30 ms	5.9 MB	cpp

```cpp
class Solution {
    public:
    bool isPalindrome(int x){
        if(x < 0 || (x % 10 == 0 && x != 0)){
            return false;
        }

        int revertedNumber = 0;
        while(x > revertedNumber) {
            revertedNumber = revertedNumber * 10 + x % 10;
            x /= 10;
        }

        return x == revertedNumber || x == revertedNumber/10;
    }
};
```

Your previous code was restored from your local storage. Reset to default

Testcase | Run Code Result | Debugger

Success Details ›

Runtime: 3 ms, faster than 16.90% of C++ online submissions for Valid Parentheses.

Memory Usage: 6.1 MB, less than 99.76% of C++ online submissions for Valid Parentheses.

Next challenges:

Generate Parentheses | Longest Valid Parentheses | Remove Invalid Parentheses | Check If Word Is Valid After Substitutions

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
09/20/2021 18:56	Accepted	3 ms	6.1 MB	cpp

```cpp
class Solution {
public:
    bool isValid(string s) {
        stack<char> st;

        for(int i = 0; i < s.length(); i++) {
            char top = st.empty() ? '#' : st.top();

            if(top == '(' && s[i] == ')') {
                st.pop();
            }
            else if(top == '{' && s[i] == '}') {
                st.pop();
            }
            else if(top == '[' && s[i] == ']') {
                st.pop();
            }
            else st.push(s[i]);
        }

        return st.empty();
    }
};
```

Testcase | Run Code Result | Debugger

Accepted Runtime: 5 ms

Success Details ›

Runtime: 12 ms, faster than 36.65% of C++ online submissions for Merge Two Sorted Lists.

Memory Usage: 14.7 MB, less than 99.51% of C++ online submissions for Merge Two Sorted Lists.

Next challenges:

Merge k Sorted Lists | Merge Sorted Array | Sort List | Shortest Word Distance II | Add Two Polynomials Represented as Linked Lists | Longest Common Subsequence Between Sorted Arrays

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
09/20/2021 18:56	Accepted	12 ms	14.7 MB	cpp

```cpp
class Solution {
public:
    ListNode *mergeTwoLists(ListNode *l1, ListNode *l2) {
        ListNode dummy{0};
        auto curr = &dummy;

        while (l1 && l2) {
            if (l1->val <= l2->val) {
                curr->next = l1;
                l1 = l1->next;
            }
            else {
                curr->next = l2;
                l2 = l2->next;
            }
            curr = curr->next;
        }
        curr->next = l1 ? l1 : l2;

        return dummy.next;
    }
};
```

Testcase | Run Code Result | Debugger

Accepted Runtime: 0 ms

Your input

```
[1,2,4]
[1,3,4]
```