C++ Programming

STL Practice #3

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Practice: Valid Parentheses

Given a string containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

- Valid: (), ()(), (()()), {}{}, ((((()))), ([]), ()[]{}, {[]}
- Invalid: (], ()), (][), (], ([)]

Practice: Valid Parentheses

```
bool isValid(string string) {
    map<char, char> mp; // mp the close of each
    mp[')'] = '(';
    mp[']'] = '[';
    mp['}'] = '{';

    stack<char> parentheses;

    for (char ch: string) {
        if (mp.count(ch)) {
            // A close one. It must match open one
            if (parentheses.empty())
                return false;

            char open = mp[ch];
            char cur_open = parentheses.top();

            if (open != cur_open)
                return false;
            parentheses.pop();
        }
        else
            parentheses.push(ch);
    }
    return parentheses.empty();
}
```

- If letter is open, add it
- If close, it must match curren open one on top of stack
- Use stack to get top easily
- Use map for shorter code
  - Otherwise if/else
- Trace
  - ([ ])

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."