Office Hours #3

```cpp
#include <stack>

bool isPalindrome(string &s) {
    stack<char> mStack;

    string temp;
    for (int i = 0; i < s.length(); i++) {
        mStack.push(s[i])
    }
    for (int i = 0; i < s.length(); i++) {
        temp += mStack.top();
        mStack.pop();
    }

    if (temp == s) return true;
    else return false;
}
```

```cpp
#include <stack>

bool matchingBrackets(string &s) {
    stack<char> ms;
    for (int i = 0; i < s.length(); i++) {
        switch (s[i]) {
        case '(':
        case '[':
        case '{':
            ms.push(s[i]);
            break;
        case ')':
            if (ms.top() == '(')
                ms.pop();
            else
                return false;
            break;
        case ']':
            if (ms.top() == '[')
                ms.pop();
            else
                return false;
            break;
        case '}'
            if (ms.top() == '{')
                ms.pop();
            else
                return false;
            break;
        }
    }

    if (!ms.empty()) return false;
    else return true;
}
```

```
// Diary Class Challenge from Lecture 7

#include <iostream>
#include <string>

class Diary {
    public:
        Diary(string name) {
            m_name = name;
        }

        virtual ~Diary() {

        }

        string getTitle() const {
            return m_name;
        }

        virtual void writeEntry(string entry) {
            m_entries += entry;
        }

        virtual string read() const {
            return m_entries;
        }

    private:
        string m_name;
        string m_entries;
};

class SecretDiary: public Diary {
    public:

        SecretDiary(string name): Diary("TOP-SECRET") {

        }

        virtual ~SecretDiary(){

        }

        virtual void writeEntry(string entry) {
            Diary::writeEntry(encode(s));           // don't touch Diary's privates!
        }

        virtual string read() {
            return decode(Diary::read());           // don't touch Diary's privates!
        }
};
```