Assignment 5

//B24CE1138

```
/*Parenthesis Checker:
Write a program using a stack for push, pop, peek, and isEmpty operations. Write isBalanced()
Function that Iterates through the input expression, Pushes opening brackets onto the stack. For
closing brackets, it checks the top of the stack for a matching opening bracket. Ensures that all
opening brackets are matched by the end of the traversal. Main Function: Accepts a string
expression from the user. Uses isBalanced() to determine if the parentheses in the expression are
balanced.
*/

#include <iostream>
using namespace std;
#define MAXSIZE 10

class Stack {
    char stack[MAXSIZE];
    int top;

public:
    Stack() {
        top = -1;
    }

    bool isFull() {
        return top == MAXSIZE - 1;
    }

    bool isEmpty() {
        return top < 0;
    }

    void push(char data) {
        if (isFull()) {
            cout << "Stack Overflow! Cannot push '" << data << "'\n";
            return;
        }
        top = top + 1;
        stack[top] = data;
    }

    void pop() {
        if (isEmpty()) {
            cout << "Stack Underflow! Cannot pop.\n";
            return;
        }
        top = top - 1;
    }

    char getTop() {
        if (isEmpty()) {
            return '\0';
        }
        return stack[top];
    }
```

```
};

// Function to check if brackets are balanced
bool isBalanced(string exp) {
    Stack s;

    for (int i = 0; i < (int)exp.length(); i++) {
        char ch = exp[i];

        // If opening bracket → push
        if (ch == '(' || ch == '{' || ch == '[') {
            s.push(ch);
        }
        // If closing bracket → match with top
        else if (ch == ')' || ch == '}' || ch == ']') {
            if (s.isEmpty()) {
                return false;
            }

            char topChar = s.getTop();

            if ((ch == ')' && topChar == '(') ||
                (ch == '}' && topChar == '{') ||
                (ch == ']' && topChar == '[')) {
                s.pop();
            }
            else {
                return false;
            }
        }
    }

    return s.isEmpty();
}

int main() {
    string expression;

    cout << "Enter expression: ";
    cin >> expression;

    if (isBalanced(expression))
        cout << "Balanced expression.\n";
    else
        cout << "Not balanced.\n";

    return 0;
}
```

OUTPUT:

```
Enter expression: (5+3)*(8-2)

Balanced expression.


=== Code Execution Successful ===
```

GITHUB: