

```

#include <iostream>
#include <stack>
#include <string>
using namespace std;

bool isOpening(char ch) {
    return ch == '(' || ch == '{' || ch == '[';
}

bool isClosing(char ch) {
    return ch == ')' || ch == '}' || ch == ']';
}

bool isMatchingPair(char opening, char closing) {
    return (opening == '(' && closing == ')') ||
           (opening == '{' && closing == '}') ||
           (opening == '[' && closing == ']');
}

bool isWellParenthesized(const string& expression) {
    stack<char> s;

    for (char ch : expression) {
        if (isOpening(ch)) {
            s.push(ch);
        } else if (isClosing(ch)) {
            if (s.empty() || !isMatchingPair(s.top(), ch)) {
                return false;
            }
            s.pop();
        }
    }

    return s.empty();
}

int main() {
    string expression;

```

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

```
if (isWellParenthesized(expression)) {  
    cout << "The expression is well-parenthesized." << endl;  
} else {  
    cout << "The expression is NOT well-parenthesized." << endl;  
}  
  
return 0;  
}
```