#include <iostream>

using namespace std;

class STACK {

public:

char str[20];

int max = 20;

int top;

STACK() {

top = -1;

}

bool isempty() {

return top == -1;

}

bool isfull() {

return top == (max - 1);

}

void display() {

if (isempty())

cout << "Stack is empty\n";

else {

for (int i = 0; i <= top; i++) {

cout << str[i] << " ";

}

cout << endl;

}

}

void push(char ch) {

if (!isfull()) {

top++;

str[top] = ch;

} else {

cout << "Stack is full, cannot push\n";

}

}

char pop() {

if (isempty()) {

cout << "Stack is empty, cannot pop\n";

return '\0';

} else {

return str[top--];

}

}

char gettop() {

if (!isempty())

return str[top];

else {

cout << "Stack is empty\n";

return '\0';

}

}

void checkparenthesis() {

cout << "Enter delimiter after expression (at the end):\n";

cout << "Enter expression: ";

cin.getline(str, max, '#');

char ch;

bool flag = false;

for (int i = 0; str[i] != '\0'; i++) {

if (str[i] == '(' || str[i] == '[' || str[i] == '{')

push(str[i]);

if (str[i] == ')' || str[i] == ']' || str[i] == '}') {

ch = pop();

if ((str[i] == ')' && ch != '(') ||

(str[i] == ']' && ch != '[') ||

(str[i] == '}' && ch != '{')) {

cout << "Not well parenthesized at index " << i << " = " << str[i] << endl;

flag = true;

break;

}

}

}

if (isempty() && !flag)

cout << "Well parenthesized\n";

else if (!isempty())

cout << "Not well parenthesized\n";

}

};

int main() {

int choice;

do {

STACK s;

s.checkparenthesis();

cout << "Do you want to continue? (1 for yes / 0 for no): ";

cin >> choice;

} while (choice != 0);

return 0;

}