// In any language program mostly syntax error occurs due to unbalancing delimiter such as

// (),{},[]. Write C++ program using stack to check whether given expression is well

// parenthesized or not.

#include<iostream>

using namespace std;

class Stack{

public:

const static int size = 100;

int top = -1;

char array[size];

void push(char x){

if(top == size-1){

cout << "Stack Overflow!!";

return;

}

array[++top] = x;

}

char pop(){

if(top == -1){

cout << "Stack Underflow!!";

return -1;

}

return array[top--];

}

void display(){

if(top == -1){

cout << "Stack is Empty";

}

cout << "Stack contains: ";

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

cout << array[i] << " ";

}

cout << endl;

}

char peek(){

return array[top];

}

bool isEmpty(){

if(top == -1){

return 1;

}else{

return 0;

}

}

};

int main(){

Stack brackets;

string s;

cout << "Enter the parenthesis to check balancing: ";

cin >> s;

bool is\_balanced = true;

for(int i = 0; i < s.size(); i++){

if(s[i] == '(' or s[i] == '{' or s[i] == '['){

brackets.push(s[i]);

}

else if(s[i] == ')' or s[i] == '}' or s[i] == ']'){

if(brackets.isEmpty()){

is\_balanced = false;

break;

}

char last\_bracket = brackets.peek();

if(s[i] == ')' and last\_bracket == '('){

brackets.pop();

}

else if(s[i] == '}' and last\_bracket == '{'){

brackets.pop();

}

else if(s[i] == ']' and last\_bracket == '['){

brackets.pop();

}

}

}

if(not brackets.isEmpty()){

is\_balanced = false;

}

if(is\_balanced)

cout << s << " is balanced." << endl;

else

cout << s << " is not balanced." << endl;

}