#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_SIZE 100

struct Stack {

char arr[MAX\_SIZE];

int top;

};

void initialize(struct Stack \*stack) {

stack->top = -1;

}

int isEmpty(struct Stack \*stack) {

return stack->top == -1;

}

int isFull(struct Stack \*stack) {

return stack->top == MAX\_SIZE - 1;

}

void push(struct Stack \*stack, char value) {

if (isFull(stack)) {

printf("Stack Overflow: Cannot push element %c\n", value);

} else {

stack->arr[++stack->top] = value;

}

}

char pop(struct Stack \*stack) {

if (isEmpty(stack)) {

return '\0'; // Return a special character to indicate stack underflow

} else {

return stack->arr[stack->top--];

}

}

int isBalanced(char expression[]) {

struct Stack stack;

initialize(&stack);

int len = strlen(expression);

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

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

push(&stack, expression[i]);

} else if (expression[i] == ')' || expression[i] == ']' || expression[i] == '}') {

if (isEmpty(&stack)) {

return 0;

}

char popped = pop(&stack);

if ((expression[i] == ')' && popped != '(') ||

(expression[i] == ']' && popped != '[') ||

(expression[i] == '}' && popped != '{')) {

return 0;

}

}

}

return isEmpty(&stack);

}

int main() {

char expression[MAX\_SIZE];

printf("Enter an expression: ");

scanf("%s", expression);

if (isBalanced(expression)) {

printf("The expression is balanced.\n");

} else {

printf("The expression is not balanced.\n");

}

return 0;

}

Output:

