ASSIGNMENT 1

Check balanced parentheses using stack in C

CSA0303 – DATA STRUCTURES FOR Problem Solving

NAME: KAVYA SHRI G

REG.NO: 192421052

AIM: To check the balanced parenthesis using stack in the C programming language.

ALGORITHM:

1. Open Dev C in the PC and initialize the program with an empty stack.
2. Get the input from the user and traverse the input expression to check the balanced parenthesis.
3. If the given character is open bracket push it into the stack and if it is a closing bracket pop out, check if it matches the corresponding open bracket.
4. If the given stack is not empty or unbalanced return false.
5. If the given stack is empty or balanced return true.

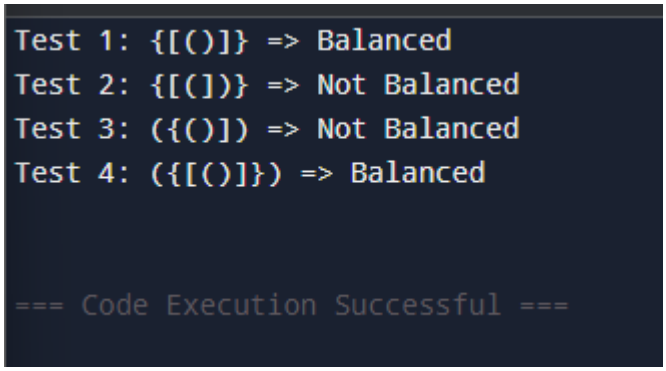
CODE:

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 100

char stack[MAX];

int top = -1;

void push(char c)

{

   if (top < MAX - 1)
```

```
    {
        stack[++top] = c;
    }
}
char pop()
{
    if (top != -1)
    {
        return stack[top--];
    }
    return '\0';
}
int isMatchingPair(char open, char close)
{
    return (open == '(' && close == ')') ||
           (open == '{' && close == '}') ||
           (open == '[' && close == ']');
}
int isBalanced(char* expr)
{
    top = -1;
    for (int i = 0; expr[i] != '\0'; i++)
    {
        if (expr[i] == '(' || expr[i] == '{' || expr[i] == '[')
        {
            push(expr[i]);
```

```
        }
        else if (expr[i] == ')' || expr[i] == '}' || expr[i] == ']')
        {
            if (top == -1 || !isMatchingPair(pop(), expr[i]))
            {
                return 0;
            }
        }
    }
    return top == -1;
}
int main()
{
    char* test1 = "{[()]}";
    char* test2 = "{[(])}";
    char* test3 = "({()])";
    char* test4 = "({[()]})";
    printf("Test 1: %s => %s\n", test1, isBalanced(test1) ? "Balanced" : "Not
Balanced");
    printf("Test 2: %s => %s\n", test2, isBalanced(test2) ? "Balanced" : "Not
Balanced");
    printf("Test 3: %s => %s\n", test3, isBalanced(test3) ? "Balanced" : "Not
Balanced");
    printf("Test 4: %s => %s\n", test4, isBalanced(test4) ? "Balanced" : "Not
Balanced");
    return 0;
}
```

OUTPUT:


Test 1: {[()]} => Balanced
Test 2: {[(])} => Not Balanced
Test 3: ({()]) => Not Balanced
Test 4: ({[()]}) => Balanced
=== Code Execution Successful ===