Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Electronics & Telecommunication Engineering
Data Structures & Algorithms Lab (DJ19ECSBL1)

Name: Nihal Nisar Shaikh **SAP ID: 60002200155** **Batch: E1-4**

Experiment no: 2 Date: 29-09-2022

Aim: Write a program in C to implement balanced parenthesis checker using STACK.

Programming Language: C

Theory: A stack is a linear data structure that follows the principle of **Last In First Out (LIFO)**. This means the last element inserted inside the stack is removed first. There are many applications of stack like reversing a string, checking balanced parenthesis, infix to postfix expression conversion etc.

- **Algorithmic steps to check balanced parenthesis:**
    - Initialize a character stack. Set top pointer of stack to -1.
    - Find length of input string using strlen function and store it in an integer variable "length".
    - Using a for loop, traverse input string from index 0 to length-1. Ignore everything you find other than the opening and the closing parenthesis
        - If the current character is a starting/opening bracket (‘(‘or ‘{‘or ‘[‘) then push it to stack.
        - If the current character is a closing bracket (‘)’ or ‘}’ or ‘]’) then pop from stack and if the popped character is the matching starting bracket, then fine else “brackets are not balanced”. If stack is empty, then input string is invalid, it means that the right parenthesis are more than left parenthesis.
    - After complete traversal, if there is some starting bracket left in stack it means left parenthesis are more than right parenthesis then “Expression is NOT BALANCED”

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Electronics & Telecommunication Engineering
Data Structures & Algorithms Lab (DJ19ECSBL1)

Name: Nihal Nisar Shaikh **SAP ID: 60002200155** **Batch: E1-4**

Code:

```
#include <iostream>
#include <string.h>

using namespace std;
char stack[20];
int top=-1;

void push (char ele)
{
    if(top==19)
    {
        cout<<"Overflow"<<endl;
    }
    top=top+1;
    stack[top]=ele;
}

char pop()
{
    if(top==-1)
    {
        cout<<"Underflow"<<endl;
    }
    top=top-1;
    return stack[top+1];
}

int match(char d,char e)
{
    if(d=='(' && e==')')
    {
        return 1;
    }
    if(d=='[' && e==']')
    {
        return 1;
    }
    if(d=='{' && e=='}')
```

Name: Nihal Nisar Shaikh **SAP ID: 60002200155** **Batch: E1-4**

```
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

int parenthesis(char exp[])
{
    for(int i=0;i<strlen(exp);i++)
    {
        if(exp[i]=='(' || exp[i]=='[' || exp[i]=='{')
        {
            push(exp[i]);
        }
        if(exp[i]==')' || exp[i]==']' || exp[i]=='}')
        {
            if(top==-1)
            {
                cout<<"Right parenthesis are more"<<endl;
                return 0;
            }
            char b=pop();
            int c=match(b,exp[i]);
            if(c==0)
            {
                cout<<"Mismatched Parenthesis"<<endl;
                return 0;
            }
        }
    }
    if(top==-1)
    {
        cout<<"Balanced parenthesis"<<endl;
        return 1;
    }
    else
    {
        cout<<"Left parenthesis are more"<<endl;
        return 0;
```

Name: Nihal Nisar Shaikh **SAP ID: 60002200155** **Batch: E1-4**

```
    }
}

int main()
{
    char exp[50];
    cin>>exp;
    int a=parenthesis(exp);
    if(a==1)
    {
        cout<<"Expression is valid"<<endl;
    }
    else
    {
        cout<<"Invalid expression"<<endl;
    }
    return 0;
}
```

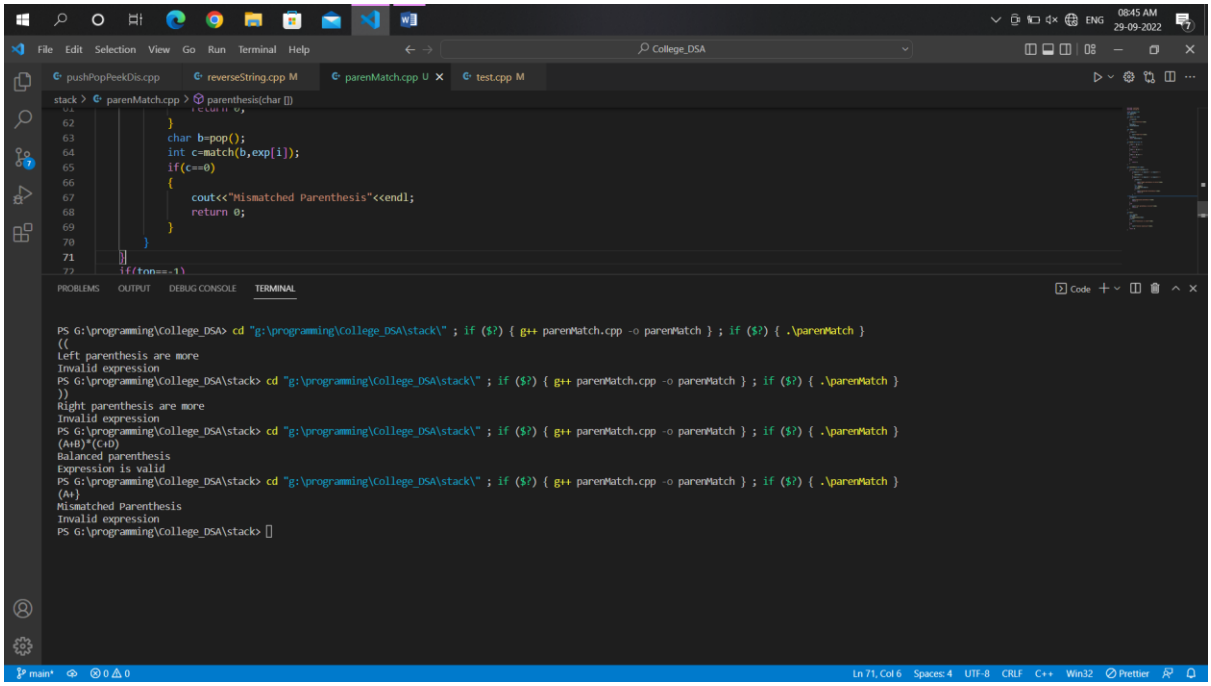
Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Electronics & Telecommunication Engineering
Data Structures & Algorithms Lab (DJ19ECSBL1)

Name: Nihal Nisar Shaikh **SAP ID: 60002200155** **Batch: E1-4**

Output:

For different combinations:



Result and Conclusion:

1) **Pushed string to stack from given input.**
2) **While pushing if parenthesis are matched then pop operation is done for similar parenthesis.**
3) **Particular output which corresponds to equation whether the equation is valid or not printed after.**
4) **Using stacks we observed whether parenthesis are matched or not for input equation.**