Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Spring, Year: 2021), B.Sc. in CSE (Day)

LAB REPORT NO: 02

Course Title: Compiler Lab

Course Code: CSE-306 **Section:** 193-DB

Lab Experiment Name: Write a C program to check whether a mathematical statement is solvable or not.

Student Details

Name	ID
Jakirul Islam	193002101

Lab Date: 05.01.2021

Course Teacher's Name: Md. Atikuzzam

Lab Report

1. TITLE OF THE LAB EXPERIMENT

Writes a C program to check whether a mathematical statement is solvable or not.

2. OBJECTIVES/AIM

1. In this program we have a input given in lab 02 requirement in classroom (a+(b+a)+C) . So, basically, we'll input requirements and it will return if the input "It is in the grammar." or "It is not in the grammar."

3. PROCEDURE / ANALYSIS / DESIGN

Declare two character arrays str[],token[] and initialize integer variables a=0,b=0,c,d.

Input the string from the user.

Repeat steps 5 to 12 till str[a] ="\0".

If str[a] =='(' or str[a] =='{' then token[b] ="4", b++.

If str[a] ==')' or str[a] =='}" then token[b] ="5", b++.

Check if isdigit(str[a]) then repeat steps 8 till isdigit(str[a])

a++.

a–, token[b] =‟6‟, b++.

If str[a]==’+‟ then token[b]=’2′,b++.

If(str[a]==’*’) then token[b]=‟3‟,b++.

a++.

token[b]=’\0′;

then print the token generated for the string .

b=0.

Repeat step 22 to 31 till token[b]!=’\0′

c=0.

Repeat step 24 to 30 till (token[b]==’6′ and token[b+1]==’2′ and token[b+2]==’6′) or

(token[b]==’6′ and token[b+1]==’3’and token[b+2]==’6′) or (token[b]==’4′ and

token[b+1]==’6′ and token[b+2]==’5′) or (token[c]!=’\0′).

token[c]=’6′;

c++;

Repeat step 27 to 28 till token[c]!=’\0′.

token[c]=token[c+2].

c++.

token[c-2]=‟\0‟.

print token.

b++.

Compare token with 6 and store the result in d.

If d=0 then print that the string is in the grammar.

Else print that the string is not in the grammar.

Stop.

Users > rudra > Desktop > C math.c > main()

```c
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>

void main() {
  int a = 0, b = 0, c;
  char str[20], tok[11];
  clrscr();
  printf("Input the expression = ");
  gets(str);
  while (str[a] != '\0') {
    if ((str[a] == '(') || (str[a] == '{')) {
      tok[b] = '4';
      b++;
    }
    if ((str[a] == ')') || (str[a] == '}')) {
      tok[b] = '5';
      b++;
    }
    if (isdigit(str[a])) {
      while (isdigit(str[a])) {
        a++;
      }
      a--;
      tok[b] = '6';
      b++;
    }
    if (str[a] == '+') {
      tok[b] = '2';
      b++;
    }
```

Output:

```
Input the expression = (a+(b+a)+C)
4242525
4242525
4242525
4242525
4242525
4242525
4242525
4242525
It is not in the grammar.
```

The output is given in the classroom is: “It is in the grammar.” but in the output, after writing the code it returns that “**It is not in the grammar**”.