

Construct a Four-Function Calculator to Evaluate Infix and Postfix Expression

Md. Jannatul Haq

ID: 2018-1-60-224

Dept. of Computer Science
and Engineering,
East West University,
Bangladesh

Sharmin Akther Rima

ID: 2018-2-60-112

Dept. of Computer Science
and Engineering,
East West University,
Bangladesh

Apu Sardar

ID: 2018-1-60-120

Dept. of Computer Science
and Engineering,
East West University,
Bangladesh

Abstract— This paper discusses how we can develop a simple calculator using the C++ language. In this project, we tried to make a calculator with four functions. This four-function are addition, subtraction, multiplication, and division. We also added some extra features along with these four functions such as squares, root, $\sin(x)$, $\cos(x)$, $\tan(x)$, and many more. We added Infix to the Postfix converter and Postfix generator in this calculator. We implemented our required code using code-blocks software.

Keywords— Infix, Postfix, Stack, Addition, Subtraction, Multiplication, Division.

I. INTRODUCTIONS

The calculators we know today were not invented until the 1970s, and the use of smartphones as calculators did not begin until at least the late nineties. But that doesn't mean that mathematical tools were not available before the 20th century- there were many different computing machines created long before digital calculators and smartphones.

In 1642, the first “calculator” was invented:

one that performed calculations through a clockwork-type of mechanism. The Pascal calculator, invented by French inventor and mathematician Blaise Pascal, was lauded for attempting arithmetic calculations previously impossible. But unfortunately, they were difficult to produce and very few were ever made. The mathematical calculator then invented by Thomas de Colmar in the mid-nineteenth century, and subsequent others, were easier to produce, but extremely large and bulky-not at all the pocket calculator we know today.

A calculator is a device with the aid using which human beings who can calculate math operations greater easily. maximum calculators have added, subtract, multiply, and divided options. Some can do rectangular roots, and a few can do the greater complicated operation. The calculator additionally can assist to do calculus and draw feature graphs. A cell phone or different laptop includes a calculator additionally which acts as an actual calculator.

The fundamental idea of this venture is that is a person-pleasant calculator. Our 4-feature calculator has no limitless capabilities, most effective implementation procedure can deliver a

character cost and true experience. This can assist others to finish greater complicated and sensible tasks in the future. We can enter infix expression in addition to postfix expression to calculate. Depending on entering, if it's miles in Infix shape, we convert the expression into Postfix first after which we generate the result with the aid of using the use of 4 capabilities operations. Again, if enter in Postfix shape we direct generate the expression the use of 4 capabilities operations. The 4 capabilities are used for addition, subtraction, multiplication, and department with the aid of using code blocks.

II. BACKGROUND STUDIES

A. Calculator Compiler Using Lex and YACC

One can enhance the easy calculator compiler with the use of Lex (Lexical Analyzer Generator) and YACC (Yet Another Compiler-Compiler). Lex continually attempts to assist writing applications whose manage glide is directed via way of means of times of normal expressions inside the enter circulation. Lex device supply is the desk of normal expressions and corresponding software fragments.

B. Logic calculator

The desk is translated to software that reads an enter circulation, copying it to an output circulation and partitioning the enter strings which suit the given expressions. On the opposite hand, the YACC device gets to enter the consumer grammar. Starting from this grammar it generates the C supply code for the parser. YACC invokes Lex to experiment with the supply code and makes use of the tokens back via way of means of Lex to construct a syntax tree. With the assistance of YACC and Lex device authors made their very own calculator compiler.

C. Common sense calculator

It is a unique device to carry out common sense operations. researchers generated a Logic Calculator with 3 operation modes: logical entailment, assessment of logical formulae, and conversion of a formulation to Disjunctive Normal Form (DNF) and Conjunctive Normal Form (CNF). It is well worth noting there isn't any present automatic equipment for those 3 computations incorporated into one device supplying a graphical calculator format.

D. Spoken 4-Function Calculator

This calculator become carried out in MATLAB. A software of MFCCs (Mel-frequency kestral coefficients) for speech-reputation functions is defined and evaluated. For education fashions, The MFCCs with lots of audio documents are extracted. To decide on spoken mathematical expression contents authors proposed the fashions. The software evaluates that expression and returns the whole equation to the consumer. Although there may be masses of room for improvement, this undertaking suggests that even easy fashions may be fairly a success in distinguishing among extra complicated words. Perhaps an extra treasured result of this undertaking become the gaining knowledge of revel in that got here in conjunction with finishing it.

E. Lexical calculator

In this calculator for lexical variables authors proposed, that while some equipment had been evolved for researchers to compute the lexical traits of words, extant assets are confined to their useability and functionality. Specifically, a few types of equipment require customers to have a few previous understandings of a few factors of the applications, and now no longer all equipment permit customers to specify their personal corpora. Additionally, modern equipment also is confined in phrases of the variety of metrics that

they can compute. To cope with those methodological gaps, this text introduces Lexical, a fast, simple, and intuitive calculator for lexical variables. Specifically, Lexical is a standalone executable that offers alternatives for customers to calculate quite a few theoretically influential surface, orthographic, phonological, and phonographic metrics for any alphabetic language, the usage of any user-targeted input, corpus file, and phonetic system. Lexical additionally comes with a fixed of well-documented Python scripts for every metric, that may be reproduced and/or changed for different studies purpose.

III. PROPOSED WORK

We all know about the precedence of operators: division and multiplication have the same priority and have higher priority than addition and subtraction, which have the same priority. A simple calculator performs most of the calculations wrong. For example, $1+2*3$ should be equal to 7 according to the normal calculator, but the problem is the simple calculator gives a result of 9. The reason a simple calculator does it all wrong is that it just multiplies the two operands, with the operator between them, and produces the result. For this wrong result, we just cannot blame the simple calculator. After all, it was meant to be simple.

So, our main goal is to rearrange the operators, to get the correct result. For that, we need to know about two more concepts in computer science: Infix expression and Postfix expression. The arithmetic expression that we understand is the Infix expression and the arithmetic expression that the computer understands is the Postfix expression. By using the algorithm of Infix to Postfix we will give the correct output for any mathematical expression.

We are working on a project to implement an arbitrary precision four-function calculator. From this project, we will create a calculator that will

calculate different types of calculations like summation, subtraction, multiplication, division, logarithm, root, percentage, and calculation of Sine, Cosine, and Tangent of a given angle in degree. We can achieve our goal from this project. We can easily perform faster calculations from this project for example we can perform some calculations (Summation, root, percentage, etc.) of the value of A and B. Overall we can say the below functions will be performed faster in our project.

A. Infix Expression

Based on entering expression we proposed a first string and a stack. One string is saved enter and any other is used for changing to postfix. Programmed scans the first string separately and if unearths any digit, the area uploads those inside the 2d string. But if the programmer unearths operators, then ships them inside the stack if the priority of the pinnacle operator of the stack is decreased then the brand-new operator goes to be delivered. Otherwise, the pinnacle detail of the stack might be delivered inside the 2D string and additionally become forth from the stack. Again, whilst the first brace will come, this detail might be driven inside the stack, and whilst cease of the primary brace will come, each operator might become forth from the stack until the primary brace might become forth After scanning the general first string, the stack will pop the relaxation of the operators from the pinnacle. We can constitute our paintings the usage of an everyday expression.

B. Postfix Expression

Based on postfix expression we proposed 4 characteristics and those are, Addition, Subtraction, Multiplication, and Division that's generated a 4-characteristic calculator. By scanning all individual postfix string with the aid of using one, the programmer converts characters to digits and push with inside the stack. If an

individual is an operator, evaluate with our 4 characteristics to decide which paintings will do.

1) Addition: The additional feature will pop digits from the pinnacle of the stack and upload those digits. Finally, push the bring about the stack.

2) Subtraction: Subtraction feature will pop digits from the pinnacle of the stack and subtract 1st popped digit from the final popped digit. Finally, push the bring about the stack.

3) Multiplication: This feature will pop digits from the pinnacle of the stack and multiply them. Finally, push the bring about the stack.

4) Division: This feature will pop digits from the pinnacle of the stack and divide if 1st digit popped digit is decreased than the final popped digit. in any other case, the incorrect equation may occur. Finally, push the bring about the stack. The result may be the pinnacle detail of the stack.

We implemented another operation two number operation. Like, Summation, Division, Subtraction, Multiplication, Square, Root, Sin, Cos, Tan, etc.

C. Software

1. Software: Codeblocks
2. Language: C++

IV. EXPERIMENTAL ANALYSIS

In this project, we made a simple calculator using C++ language. This project worked perfectly. When we want to open this project the Code Block firstly searches the file named calculator main.cpp and then enters the following code. This project reduces the time and space complexity. In the worst case, the space complexity is always $O(1)$. But in the worst case, the time complexity is $O(n)$. Here n is the length of the input expression.

V. EXPERIMENTAL RESULT

After opening this program, the program will ask the user what they want to do. Let's see the result of this program.

```
*****
                        CALCULATOR
*****

Enter The Method Which You Want to do:
    1. Equation Method      2. Two Digit Method
Enter Your Option: _

*****
                        CALCULATOR
*****

Enter The Method Which You Want to do:
    1. Equation Method      2. Two Digit Method
Enter Your Option: 1

Enter The InFix Equation: 40 + 5 + 6 - 10 * ( 6 * 5 )

                        Result: -249

Do you want to continue??? y/n: _

Do you want to continue??? y/n: y

Enter The Method Which You Want to do:
    1. Equation Method      2. Two Digit Method
Enter Your Option: 2
Please Choose Your Option:
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Squares
    6. Root
    7. Sin(X)
    8. Cos(X)
    9. Tan(X)
    10. exit

Choice:
```

```

Enter The Method Which You Want to do:
    1. Equation Method    2. Two Digit Method
Enter Your Option: 2
Please Choose Your Option:
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Squares
    6. Root
    7. Sin(X)
    8. Cos(X)
    9. Tan(X)
    10. exit

Choice: 1
Enter Two Numbers:
40 50
Sum = 90
Do you want to continue??? y/n: _

```

```

Enter The Method Which You Want to do:
    1. Equation Method    2. Two Digit Method
Enter Your Option: 2
Please Choose Your Option:
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Squares
    6. Root
    7. Sin(X)
    8. Cos(X)
    9. Tan(X)
    10. exit

Choice: 9
Enter any Angle: 45
Tan(45)= 1
Do you want to continue??? y/n: _

```

VI. CONCLUSION

This calculator is totally user-friendly. In our project, we tried our best to solve all the bugs that occurred during designing this code. Hence, in conclusion, we were successful to make the code fully functional without much of an error.

This project is done with only C++ language. In this project, we can simply use it for calculating basic mathematical operations like summation, division, multiplication, tan, cos, sin, etc. But in our project, you cannot calculate any binary number operation this is the main restriction of our project. So, overall, this was our project work.

VII. REFERENCES

1. <https://www.semanticscholar.org/paper/On-the-development-of-a-logic-calculator%3A-a-novel-Ch%C3%A1vez-Bosquez-Parra/22e0ab32bfb8231215465c07ab1b35da5d6d857f>
2. https://www.researchgate.net/publication/351253364_LexiCAL_A_calculator_for_lexical_variable_s
3. <https://www.oool.co.uk/blog/brief-history-calculator/> [Last Access: 28-04-2022]
4. <https://www.freecodecamp.org/news/zipline-build-a-smart-javascript-calculator-65e21a8b0b0c/> [Last Access: 06-05-2022]
5. <https://edtechmagazine.com/k12/article/2012/11/calculating-firsts-visual-history-calculators>
6. <https://www.schoolmart.com/2017/03/29/the-calculator-a-brief-history/> [Last Access: 05-05-2022]