**IdeagenCalculator Summary ([Github](https://github.com/fadhilnor/IdeagenCalculator))**

Required feature = 2 hours

Bonus feature = 3 hours

Documentation = 1 hour

**Total time = 6 hours**

Approach 1

Simple way to do it is by using DataTable, which supports basic calculation but has certain limitation and does not support nested expression.

using System.Data;

public static double Calculate(string sum)

{

DataTable calculationTable = new DataTable();

return calculationTable.Compute(sum, "");

}

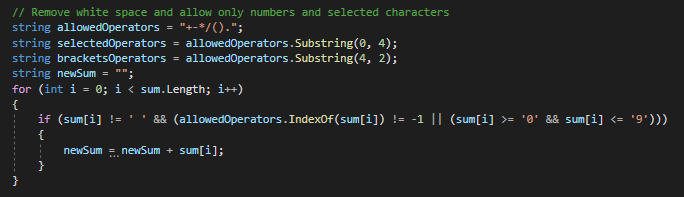
Approach 2

Second approach is by breaking the expression down using logic and pattern. A simple expression should consist of three variables:

1. Numbers. (In this case with decimal values)
2. Operators (+, -, \*, /)
3. Brackets (Optional - For nested and multiplication)

Although stated that the function only accept numbers, it’s recommended to filter any unwanted characters besides the three variables as well as removing any empty spaces.

The filter is shown here:



Example 1

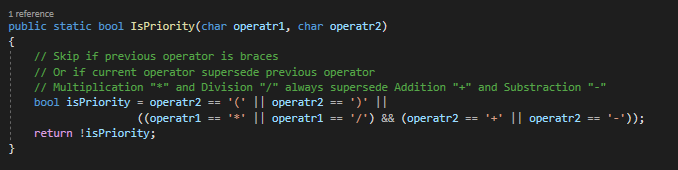
Given the expression:

*1 + 1 \* 3*

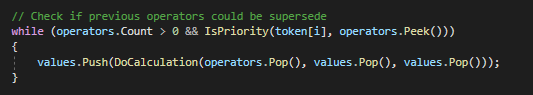
Although the normal way to solve it is from left to right, but since the multiplication supersede addition, it should be solve the multiplication first and then addition.

*1 + 1 \* 3 = 1 +* ***4 = 5***

This function determines their priority:



Which is being called here:



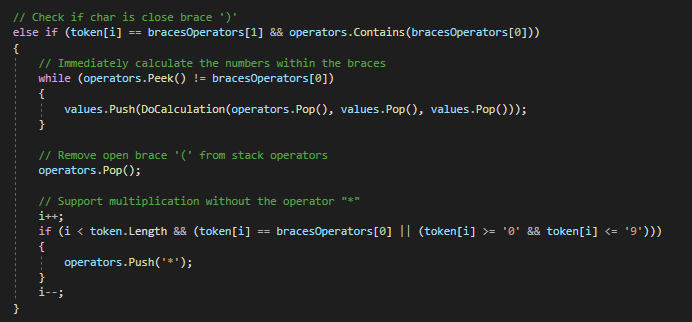
Example 2

Given the expression:

*( 11.5 + 15.4 ) \* 10.1*

Any expression within the brackets “(…)”, should be executed before doing other calculation and followed by executing the remaining calculation.

*( 11.5 + 15.4 ) \* 10.1 = (* ***26.9*** *) \* 10.1 =* ***271.69***

This condition will be executed when a close bracket is found ‘)’:

Example 3

Given the expression:

*10 - ( 2 + 3 \* ( 7 - 5 ) )*

As with the previous example, any expression that is within brackets “(…)” will be calculated first. Then priority operator (multiplier) will be calculated followed by addition.

*10 - ( 2 + 3 \* ( 7 - 5 ) ) = 10 - ( 2 + 3 \* (* ***2*** *) ) = 10 - ( 2 +* ***6*** *) = 10 – (****8****) =* ***2***