이영진, 박성배, 배재영 기초프로그래밍 팀 과제

20181277 이영진 20180353 박성배 20180203 배재영

Calculator Program

[Features]

-you can type in expressions with multiple operators.

-this calculator follows the Arithmetic operation precedence

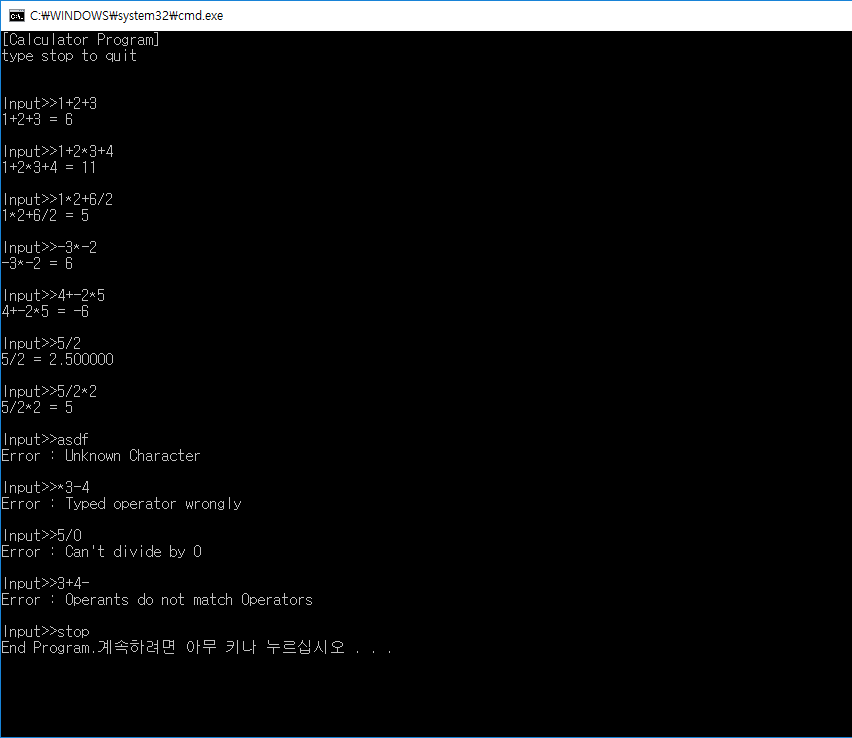
-you can use negative numbers.

-the calculating process will be done with float-type. If the result is an integer-type, it will print that way.

-you are not allowed to input any character except numbers and operators(+, -, \*, /)

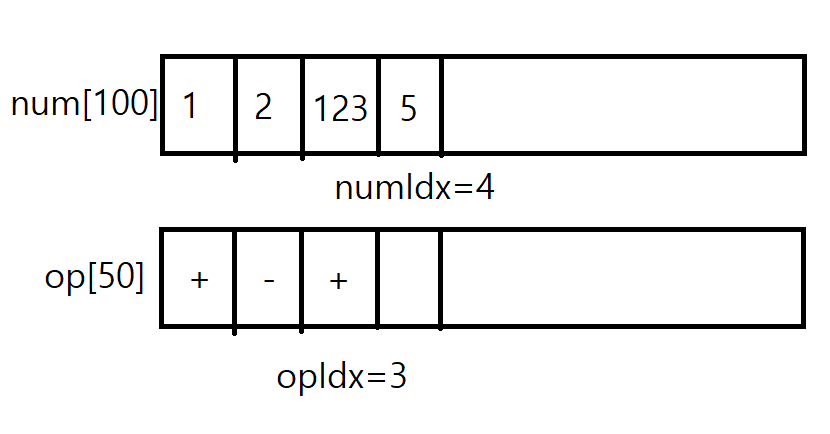
-you cannot use brackets.

-If you did something wrong, an error message will occur.



[Code structure]

The program uses an array of numbers and an array of operators.



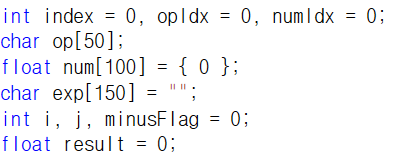
Then, it calculates going back and forth between these arrays.

[Code description]

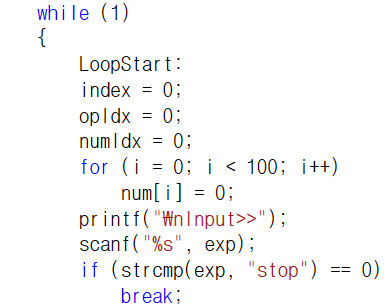


isNum : determines if the character is an number or not. If number, returns 1. If not, 0

isOp : determines if the character is an operator or not. If number, returns 1. If not, 0



All declarations needed.

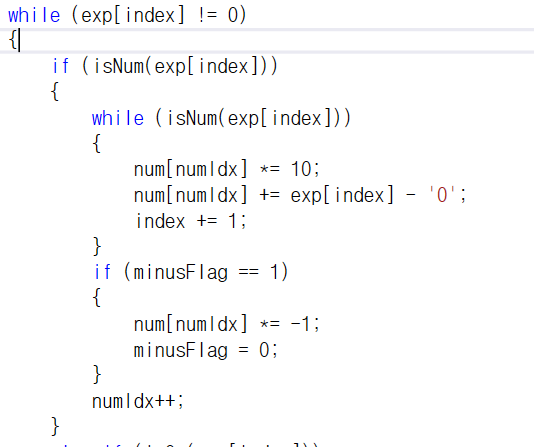


This programs uses a big loop to do multiple calculations.

The label LoopStart is where the program goes if it meets an error.

Under that, all initializations.

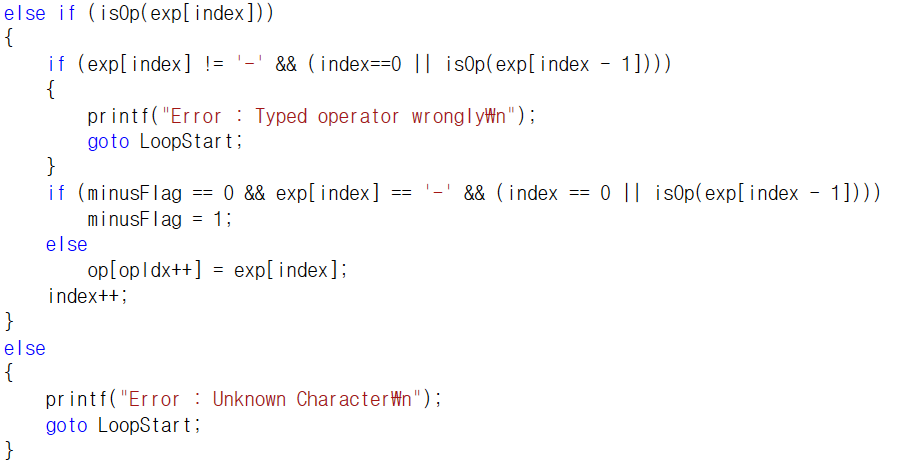
Then, checks if input string was “stop”. If it was, get out of the loop.



Goes through each character in the input string.

If a character is a number, save it in the num array. If there are more numbers, it means it is one number (if 12, the array will be {1,2}), so add it in the same spot.

minusFlag is for later, but it makes the number negative.



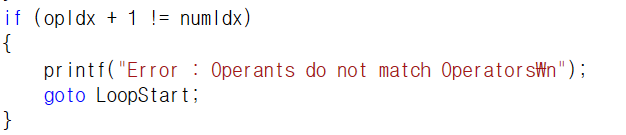
If a character is an operator(not -),

And it is the first character or the character before is also an operator, print an error message and restart the loop.

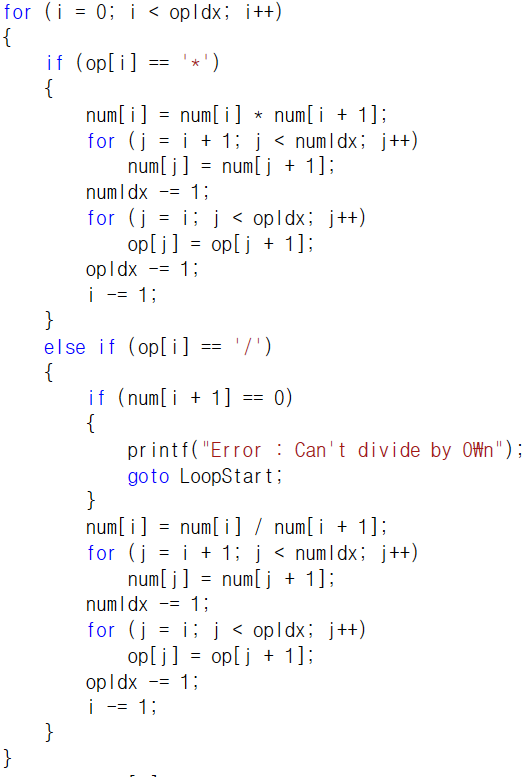
If it is a – operator and the character before is an operator, minusFlag is 1. This means the number is negative. Ex) if 2+-3, -operator is before another operator so the next number is negative. Also if 2---3, the number will be treated negative twice. This is an error. So I added the condition minusFlag==0.

If it doesn’t apply to any of these conditions, just add it to the op array. The things above is just for error and negative numbers.

If a character is neither a number or an operator, print an error.



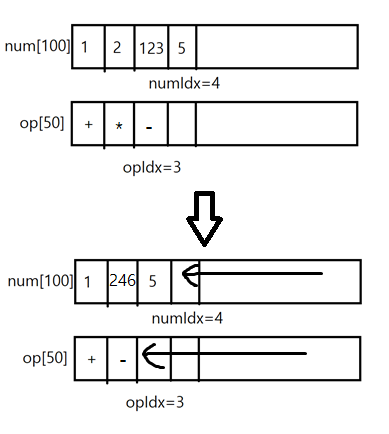
If it was did right, the number of numbers will be one more than the number of operators. If not, print an error.

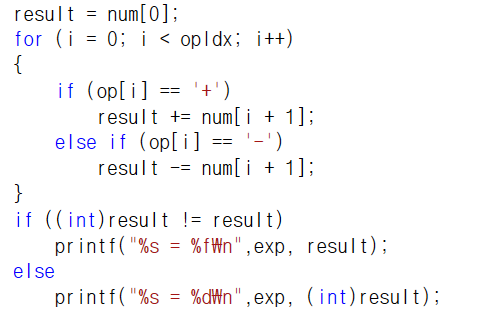


Then do the multiply and divide operator first.

If the operator is one of those, change the number at the position and push all other indices.

The structure will be like this.





Then, do the rest of the calculation.

If the casted result and the original result is different, it means its value is a float-type.

If not, print it as an integer-type.