Lab 07 - Expression Format

Direction: Submit the modified cpp file in the Labs directory of your github repository and/or as an attachment on Google classroom under the Lab07 assessment. The submission must be modified cpp file.

Complete the following objectives

A NWI computer has 20-bit words and instructions in the format

```
[opcode|operandX|operandY]
```

where opcode is 4 bits and each operand is a memory reference that is 8 bits. The instruction commands list for the computer are

Opcode	Description
0	Halts the program
1	Adds the contents of M(X) and M(Y) and puts the result in X
2	Subtracts the contents of M(Y) from M(X) and puts the result in X
3	Multiples the contents of M(X) and M(Y) and puts the least significant bits of the result in X and most significant bits in AC
4	Divides the contents of M(X) by M(Y) and stores the quotient in X and the remainder in AC
5	Transfers M(X) to AC
6	Transfers contents from AC to X
7	Transfers M(X) to Y
8	Takes next instruction from M(X)
9	If AC >= 0, takes next instrution from M(X)
A	If AC >= 0, takes next instrution from M(X); otherwise, from M(Y)

where instructions that work with a single operand sets the second operand to 00

1. Using the information above, copy the link

   https://forms.gle/zQecSQDGFwThi97j8

   to your browser, and complete the form.

2. Computers work on either postfix or prefix arithmetic expression. Write a string function named `ToPrefix()` whose header is

   ```
   string ToPrefix(string exp)
   ```

   If *exp* is a valid simple numerical infix expression (one operation) string whose operands are lowercase letters, the function converts *exp* to a prefix expression string; otherwise, it returns an empty string. For instance, the calls `ToPrefix("a + b")`and `ToPrefix("a * b + c")` will return `"+ a b"` and `""` respectively. There must be a space between operands and operation in the output.

   Hint: store all non-space characters then check them.