Assignment 1 – A file based calculator

In this assignment you are going to create a calculator that reads in commands from an input file called `input.txt` and performs the actions required by the commands.

The program needs to consist of multiple functions. You need your main and at least two additional functions. You can have more than two if desired.

You cannot use any global variables in your programs except for const values. All data that is needed by a called function will either have to already be part of that function or it will have to be passed as a parameter.

If you need to pass in **ifstream** to a function you need to pass it as a non-const reference. The compiler will give you an error message if you try to pass an **ifstream** by copy. For example:

```
int exampleFunction(ifstream &inputFileStream)
{
      int val;
       inputFileStream >> val;
      return val;
}
```

The general syntax of a commands are as follows:

```
      action number
      action
      action text
```

The input command may contain whitespace before the action value. There must be whitespace separating action from number or from text. The text option is only used for comments (below). Anything after the number before the newline is ignored. If the command consists of only the action any text after action before the newline is ignored.

Here are the commands that must be supported by the calculator:

Command:	Meaning
`# comment`	The entire line of input is ignored
`= number`	Assign value of number to the "accumulator"
`->`	Sore the accumulator value to the calculator's memory

`<-`	Set the accumulator to the current value in memory
`+ number`	Add number to the value in the accumulator and put the new value in the accumulator
`- number`	Subtract number from the value in the accumulator and save the new value in the accumulator
`* number`	Similar to + and - but with multiplication
`/ number`	Divide the accumulator by number with the result stored back to the accumulator. Output an error message if the number is 0.
`<<`	Display the contents of the accumulator to cout with 5 digits to the right of the decimal point
`s`	Take the square root of the accumulator with the result replacing the accumulator
`p number`	Take accumulator to the number value, pow(accumulator, number), and update the accumulator with the result

Note that the actions here are +m +M -m -M etc. Note there is no whitespace between the + or - * or / and the m or M. Either m or M will indicate the calculators one memory variable.

Command	Meaning
`+m or +M`	Add memory to accumulator
`-m or -M`	Subtract memory from accumulator
`*m or *M`	Multiply accumulator by memory
`/m or /M`	Divide accumulator by memory. Output a message if division by zero is **attempted.**

Other information about the project:

Set accumulator and memory to 0.0 at start.

All input numbers are treated as double values. You can assume valid numbers have been specified.

The program will read commands from a file called

`input.txt`

Output an error message if any action other than the above is read in from the input file. Ignore the rest of the input line for an invalid action.

Output an error message if the file cannot be opened and end the program.

Here is a sample `input.txt` file:

```
# This is an example of some of the functions of the file based
calculator
# Set the accumulator
= 144
# Take the square root and output the value
S
<<
# store the value in memory
->
# calculate memory * memory
*m
# output memory squared. This should be the original 144
<<
```

Here is some sample output from the above:

```
Result is 12.00000
Result is 144.00000
```

Note that the above sample does not test every command used by the calculator. Make sure you do a thorough job of testing your commands.

Submit your source file with the name `assignment1.cpp`. Use meaningful variable names and use comments throughout your code. Put a block of comments at the beginning with a summary of what the program does. Also include your name in this block of comments. Make sure you use good indentation and formatting for your program.