1. The basic idea for the code, is first read the input character by character, then do the arithmetic, finally print the number with 4 digits precision.

   When reading the input, we keep tracking whether the input is a operator (i.e $+-*/.$) or a number.

   If it is a number, then we track if it is the fractional part or integer part. In order to do so we create a boolean variable $fraction$. If we encounter a dot we set $fraction$ to true (note: we use 1 for true -1 for false). We also have a boolean variable called negative in order to track whether the next operand is negative or not.

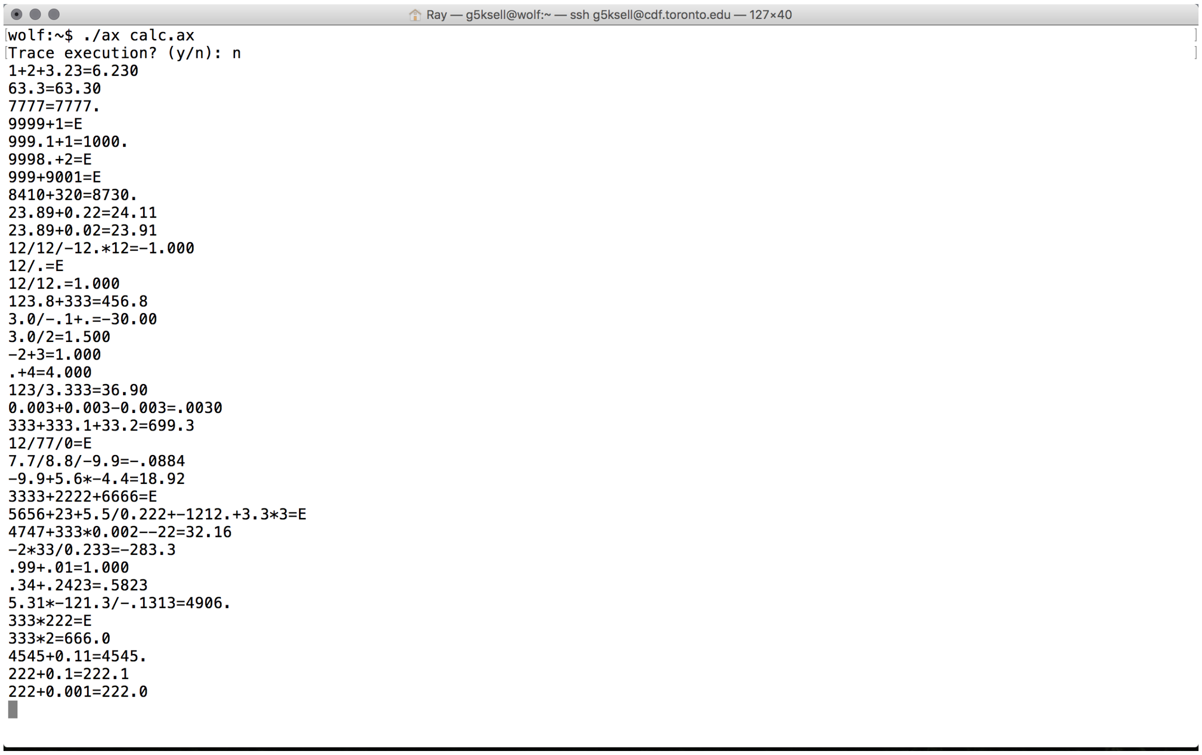
   If the input is an operator $+-*/$, we check whether the previous input is a number or dot. If it is a number or a dot then all operators are valid and we do the corresponding operation. If it is not a number then only $-$ is valid, which means next operand is negative. Otherwise, ignore the input and brunch to read input.

   If the input is $=$ then we check if the previous input is a number or dot. If it is then $=$ is a valid input, otherwise ignore the input and brunch to read input. After we received $=$ then the only valid input would be *esc*, and if the next input is *esc*, we brunch to print result otherwise, ignore and wait for the next input. If we received a *esc* input and the previous input is not $=$, then just ignore and wait for next input.

   For printing part, we are basically using the idea from the course website for PRINTINT, with a slight modification. For printing a floating number we just keep tracing the digits of the integer part and decimal position.

   After we print the result we brunch back to the beginning of the program and waiting for the next computation.

code and sample output are attached with more detailed comment



```
wolf:~$ ./ax calc.ax
Trace execution? (y/n): n
1+2+3.23=6.230
63.3=63.30
7777=7777.
9999+1=E
999.1+1=1000.
9998.+2=E
999+9001=E
8410+320=8730.
23.89+0.22=24.11
23.89+0.02=23.91
12/12/-12.*12=-1.000
12/.=E
12/12.=1.000
123.8+333=456.8
3.0/-.1+.=-30.00
3.0/2=1.500
-2+3=1.000
.+4=4.000
123/3.333=36.90
0.003+0.003-0.003=.0030
333+333.1+33.2=699.3
12/77/0=E
7.7/8.8/-9.9=-.0884
-9.9+5.6*-4.4=18.92
3333+2222+6666=E
5656+23+5.5/0.222+-1212.+3.3*3=E
4747+333*0.002--22=32.16
-2*33/0.233=-283.3
.99+.01=1.000
.34+.2423=.5823
5.31*-121.3/-.1313=4906.
333*222=E
333*2=666.0
4545+0.11=4545.
222+0.1=222.1
222+0.001=222.0
```

2. (a) BNT m

$$\begin{aligned} AC &\rightarrow AL1 \\ one &\rightarrow AL2 \\ IR &\rightarrow ALF \\ ALU &\rightarrow AC \\ IR &\rightarrow m0 \\ PC+1 &\rightarrow m1 \\ AC &\rightarrow MS \\ MUX &\rightarrow PC \end{aligned}$$

(b) EXE m

$$\begin{aligned} IR &\rightarrow MAR \\ RAM &\rightarrow IR \end{aligned}$$