day34 frequency analyzer
Due Monday 11/2/20 by midnight

Open the repl attached to this assignment in the Google classroom. It contains the following program:

```
text="""3.1415926535"""
numbers='0123456789'
count=[]

for x in numbers:
    num=text.count(x)
    count.append(num)
print(count)
```

Run the code and Python prints out the following:

```
[0, 2, 1, 2, 1, 3, 1, 0, 0, 1]
```

This is the number of times each digit (0 through 9) shows up in the first 10 digits of pi. That is, in "3.1415926" there are no 0s, two 1s, one 2, etc.

Here's how the program works:

> The first three lines set up your variables (the variable **text** contains the stuff we're analyzing, **numbers** contains the characters we're counting, and **count** is an empty list where we'll put our results.)
>
> The "for x in numbers" tells Python to loop repeatedly, putting one character at a time into **x** from the variable **numbers**.
>
> num=text.count(x) asks Python to count how many times the current character (x) shows up in the variable text.
>
> We append num to the list to save the results.
>
> After the for loop we print the list.

Look at the code, read my explanation, and work to understand what each part is doing. If you don't understand what the parts are doing, you will have a hard time doing the rest of the assignment. If you're unclear on what is happening you might take a few minutes to watch my video where I talk you through it all.

Also before you move on, add some digits to the data inside the triple quotes. For example, add your phone number, or the number 7 a bunch of times, something that will allow you to see a different output. Run the program again. Does it count the numbers you added properly?

For example, if I change my variable to the following:

```
text="""3.14159265357777777"""
```

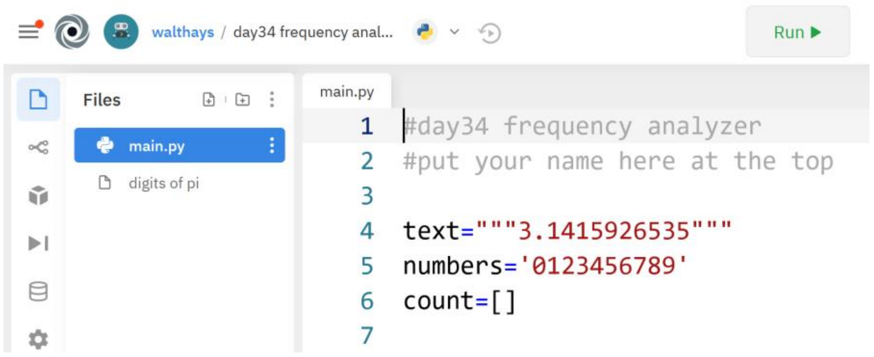
…I get this output:

```
[0, 2, 1, 2, 1, 3, 1, 8, 0, 1]
```

The number 8 over in the output is for the 8 number 7s that are now part of the variable "text". When it runs, Python counts them all and reports them in the output.

(Continued on the next page.)

We're now going to process a larger piece of text. Over on the left in repl there is a listing that shows you two files in today's project: main.py and digits of pi:



```
#day34 frequency analyzer
#put your name here at the top

text="""3.1415926535"""
numbers='0123456789'
count=[]

```

Click on the "digits of pi" listing on the left, then do a control-A to select all, then control-C to copy, then switch back to main.py, highlight the line `text="""3.1415926535"""` and then do control-V to paste.

All that stuff you just pasted in are over 2000 digits of pi. (Pi is related to Geometry and the area and circumference of circles, and it is an irrational number, a number which goes on forever, never repeating. There’s a web site that has over a million digits of pi, believe it or not!)

Now run the program. You should get the following:

```
[215, 244, 261, 221, 251, 264, 256, 237, 257, 270]
```

215 is how many times the digit "0" shows up in the first 2476 digits of pi, 244 is how many 1s there were, etc.

Today's assignment is to make this information print out in a readable way. I want you to print the info out as follows:

```
Analysis of the digits of pi:

#    count   %
0    215      8.68%
1    244      9.85%
2    261     10.54%
3    221      8.93%
4    251     10.14%
5    264     10.66%
6    256     10.34%
7    237      9.57%
8    257     10.38%
9    270      10.9%
2476 digits of pi processed
```

I walk you through today's assignment on the next pages.

Here we go!

First, remove the line that says "`print(count)`".

Then, below the existing program add this line:

```
n = 0
```

Then these lines:

```
for x in count:
   print(n,str(x))
   n=n+1
```

Run the program. It should print out the following:

```
0 215
1 244
2 261
3 221
4 251
5 264
6 256
7 237
8 257
9 270
```

Now, change the `print(n,str(x))` line to be this:

```
print(n,str(x).rjust(8))
```

Now your code prints this:

```
0      215
1      244
2      261
3      221
4      251
5      264
6      256
7      237
8      257
9      270
```

This is because rjust(8) prints some spaces to the left of the count value.

You can read line by line the following: "In the numbers we analyzed, the digit 0 occurred 215 times, the digit 1 occurred 244 times, etc…"

Let's add a header. Add this before the for loop:

```
print("#      count   %")
```

(That's 6 spaces between # and count and 3 spaces between count and %.)

Next, back inside the for loop, **change your print line** to this (you are just adding **,end=""** to the end):

```
print(n,str(x).rjust(8),end="")
```

(continued on next page)

After that print line in the for loop add the following lines:

```
percentage = x/sum(count)*100
percentage = round(percentage, 2)
print(str(percentage).rjust(9)+"%")
```

These calculate the percentage times that each digit shows up, then rounds it to two decimals, then prints it. You should now have the following:

```
#      count   %
0      215     8.68%
1      244     9.85%
2      261    10.54%
3      221     8.93%
4      251    10.14%
5      264    10.66%
6      256    10.34%
7      237     9.57%
8      257    10.38%
9      270     10.9%
```

The last thing is to add a line that says how many digits were processed. Add this:

```
print(sum(count),"digits processed")
```

This prints the total number of digits that the program analyzed.

I made a video where I walk you through doing this whole thing. That might help if you get stuck. Here is a link.