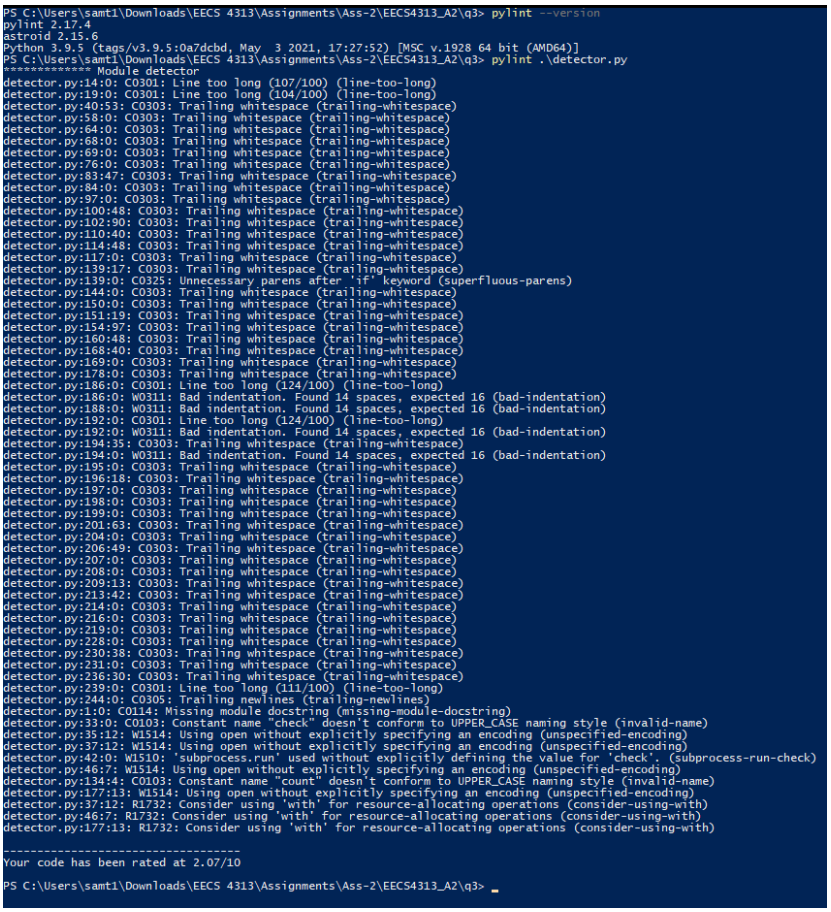
Question 3



This is the first output on the program when pylint is run. Here we see that the score is 2.07/10 according to it.

The most interesting bugs that I found in my program were:

1. On line 42 we see the subprocess-run-check issue where I don't use a check variable as a Boolean value to see if the process's output was correct or not. I blindly move forward after running the command without testing if it ran with or without errors.
2. On line 139 we see the Superfluous – parens issue which is there because python can write conditional statements without parenthesis and this behavior is not observed in Java and other languages a lot. Having parenthesis doesn't cause the code to break but still not considered good practice.

Other bugs present mostly were related to trailing spaces or indentations in the code. As Python is an indent – based language it is very important to indent after any conditions if there is need for a sub-condition. There were some naming errors as well which represent the proper styling required for any specific type of variables used, for example Boolean. Pylint was correct in informing about the subprocess issue, but it wasn't really needed as by default the value is false for check. The parenthesis was a correct observation by pylint to inform it wasn't needed at all.

Lesson Learned is to stick to proper programming conventions according to each different language and also look into a method/function before using it and see if it needs some mandatory parameters.

RUN 2

```
PS C:\Users\samt1\Downloads\EECS 4313\Assignments\Ass-2\EECS4313_A2\q3> pylint .\detector-RUN2.py
************* Module detector-RUN2
detector-RUN2.py:15:0: C0301: Line too long (107/100) (line-too-long)
detector-RUN2.py:20:0: C0301: Line too long (104/100) (line-too-long)
detector-RUN2.py:186:0: C0301: Line too long (126/100) (line-too-long)
detector-RUN2.py:192:0: C0301: Line too long (126/100) (line-too-long)
detector-RUN2.py:238:0: C0304: Final newline missing (missing-final-newline)
detector-RUN2.py:238:0: C0301: Line too long (111/100) (line-too-long)
detector-RUN2.py:1:0: C0103: Module name "detector-RUN2" doesn't conform to snake_case naming style (invalid-name)
detector-RUN2.py:36:12: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
detector-RUN2.py:38:12: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
detector-RUN2.py:47:7: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
detector-RUN2.py:177:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
detector-RUN2.py:38:12: R1732: Consider using 'with' for resource-allocating operations (consider-using-with)
detector-RUN2.py:47:7: R1732: Consider using 'with' for resource-allocating operations (consider-using-with)
detector-RUN2.py:177:13: R1732: Consider using 'with' for resource-allocating operations (consider-using-with)

-----------------------------------
Your code has been rated at 8.29/10

PS C:\Users\samt1\Downloads\EECS 4313\Assignments\Ass-2\EECS4313_A2\q3>
```

After Fixing most of the whitespace and indentation issues, with some constant naming conventions being followed correctly the code's RUN2 version is able to reach 8.29/10

Some issues still persist like line-too-long, and a new issue also appeared here where I accidentally removed the last new line.

Another one is naming for the document is not up to a particular style.

Some of these errors are not really required as the code takes care of them, because they are only warnings. So, I simply disable pylint from detecting them.

RUN 3

```
PS C:\Users\samt1\Downloads\EECS 4313\Assignments\Ass-2\EECS4313_A2\q3> pylint .\Run3.py
************* Module Run3
Run3.py:1:0: C0103: Module name "Run3" doesn't conform to snake_case naming style (invalid-name)

-----------------------------------
Your code has been rated at 9.88/10

PS C:\Users\samt1\Downloads\EECS 4313\Assignments\Ass-2\EECS4313_A2\q3>
```

On run 3 we see only the file name is not unto the required standard and needs to be changed to something better.

RUN 4

```
PS C:\Users\samt1\Downloads\EECS 4313\Assignments\Ass-2\EECS4313_A2\q3> pylint .\final.py

-----------------------------------
Your code has been rated at 10.00/10

PS C:\Users\samt1\Downloads\EECS 4313\Assignments\Ass-2\EECS4313_A2\q3>
```

As you can see in the final run, we have a perfect 10/10 from pylint and now we are able to confirm the code is up to standards.