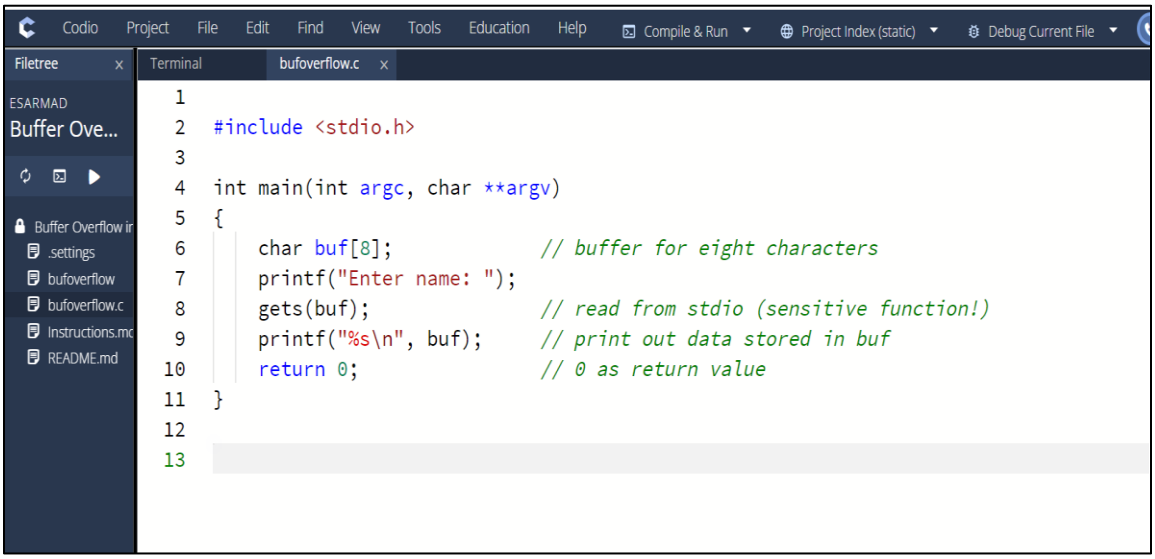
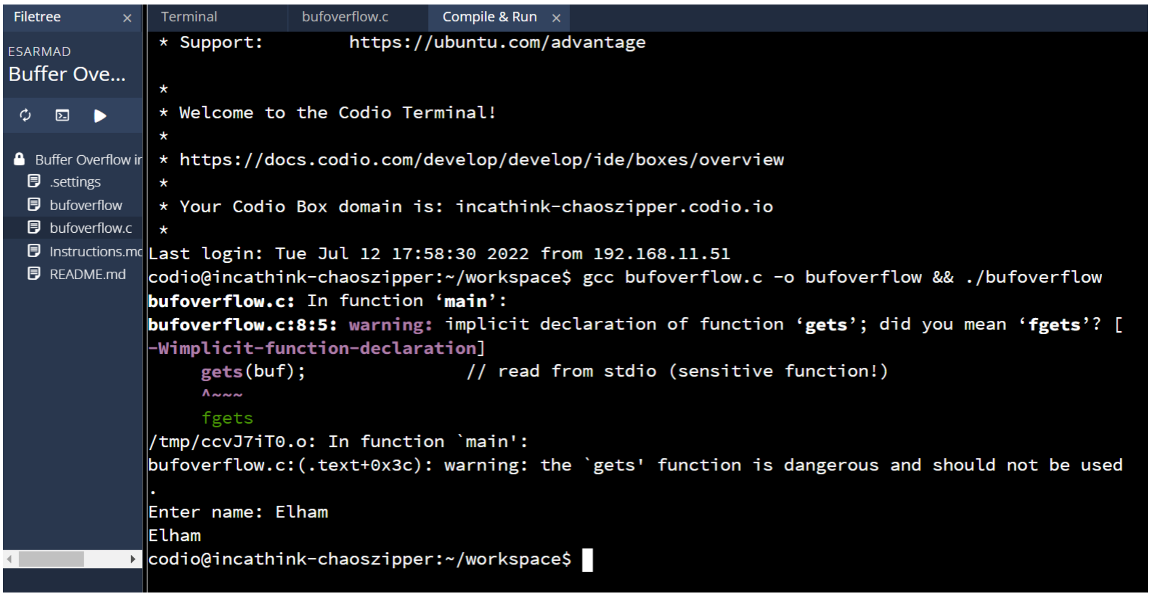
Exploring Python Tools and Features

Part 1: a simple program written in C language that creates a buffer and then asks you for a name, and prints it back out to the screen:



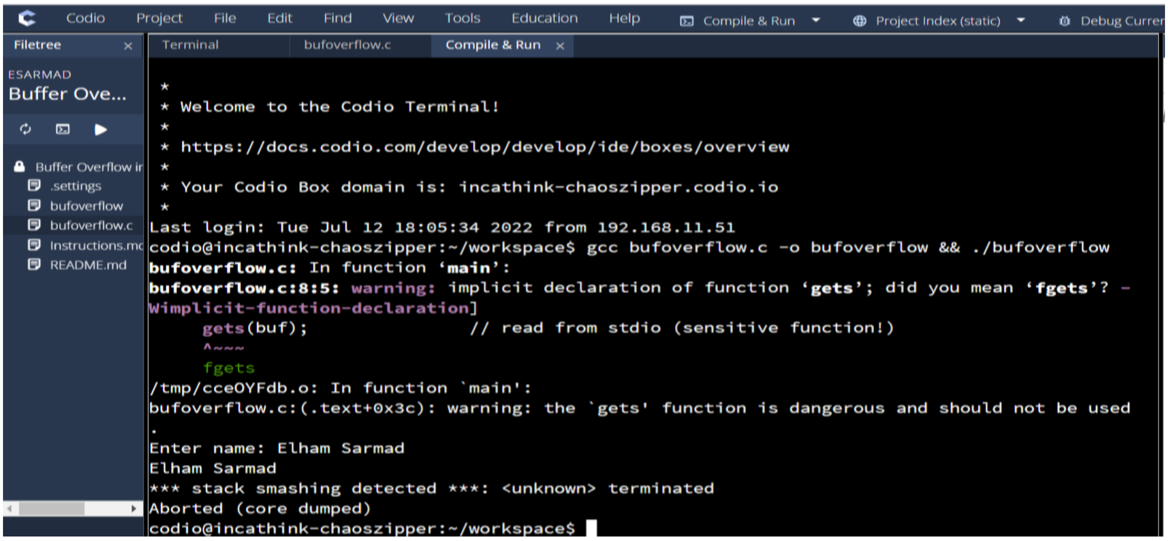
```

#include <stdio.h>

int main(int argc, char **argv)
{
    char buf[8];            // buffer for eight characters
    printf("Enter name: ");
    gets(buf);              // read from stdio (sensitive function!)
    printf("%s\n", buf);    // print out data stored in buf
    return 0;               // 0 as return value
}


```

First run:



```
 * Support:        https://ubuntu.com/advantage

 *
 * Welcome to the Codio Terminal!
 *
 * https://docs.codio.com/develop/develop/ide/boxes/overview
 *
 * Your Codio Box domain is: incathink-chaoszipper.codio.io
 *
Last login: Tue Jul 12 17:58:30 2022 from 192.168.11.51
codio@incathink-chaoszipper:~/workspace$ gcc bufoverflow.c -o bufoverflow && ./bufoverflow
bufoverflow.c: In function ‘main’:
bufoverflow.c:8:5: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [
-Wimplicit-function-declaration]
     gets(buf);              // read from stdio (sensitive function!)
     ^~~~
     fgets
/tmp/ccvJ7iT0.o: In function `main':
bufoverflow.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used
.
Enter name: Elham
Elham
codio@incathink-chaoszipper:~/workspace$ 
```

Second run:



```
 *
 * Welcome to the Codio Terminal!
 *
 * https://docs.codio.com/develop/develop/ide/boxes/overview
 *
 * Your Codio Box domain is: incathink-chaoszipper.codio.io
 *
Last login: Tue Jul 12 18:05:34 2022 from 192.168.11.51
codio@incathink-chaoszipper:~/workspace$ gcc bufoverflow.c -o bufoverflow && ./bufoverflow
bufoverflow.c: In function ‘main’:
bufoverflow.c:8:5: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? -
Wimplicit-function-declaration]
     gets(buf);                // read from stdio (sensitive function!)
     ^~~~
     fgets
/tmp/cceOYFdb.o: In function `main':
bufoverflow.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used
.
Enter name: Elham Sarmad
Elham Sarmad
*** stack smashing detected ***: <unknown> terminated
Aborted (core dumped)
codio@incathink-chaoszipper:~/workspace$
```

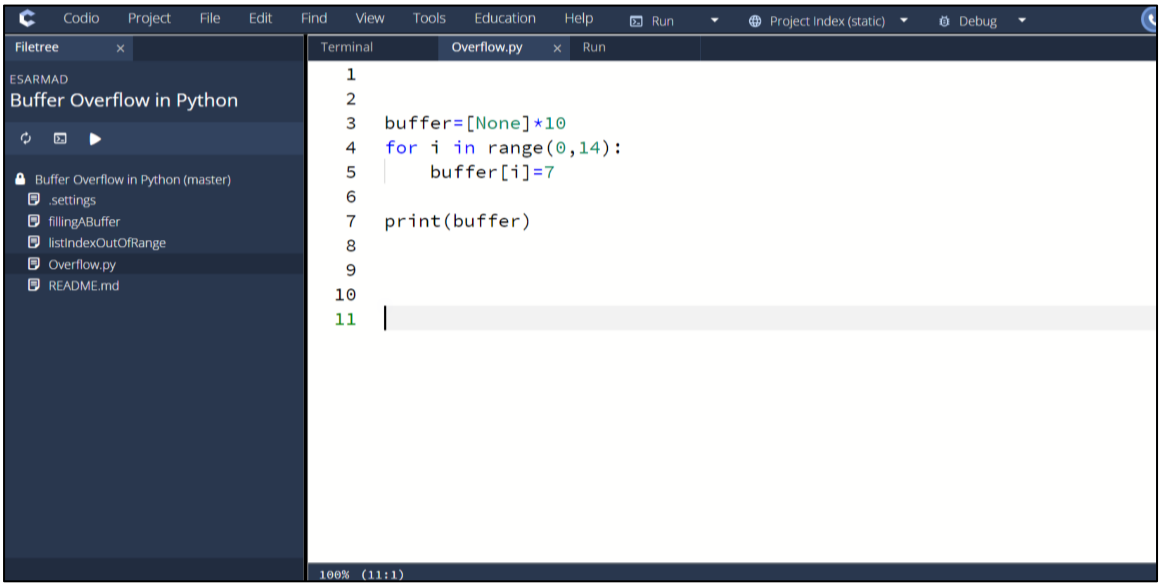
- ### What happens?

  The compiler throws “Stack smashing detected” error.

- ### What does the output message mean?

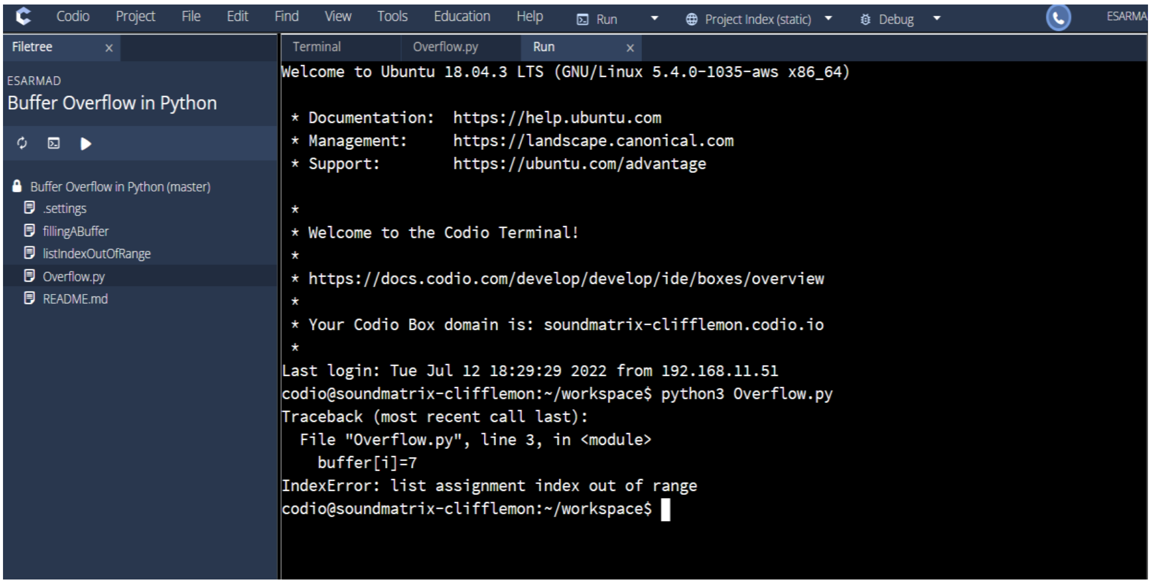
  It means that buffer overflow has occurred. In other words, the user’s input exceeds the capacity of the buffer.

Part 2:



```


buffer=[None]*10
for i in range(0,14):
    buffer[i]=7

print(buffer)




```

Run:



```
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.4.0-1035-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 *
 * Welcome to the Codio Terminal!
 *
 * https://docs.codio.com/develop/develop/ide/boxes/overview
 *
 * Your Codio Box domain is: soundmatrix-clifflemon.codio.io
 *
Last login: Tue Jul 12 18:29:29 2022 from 192.168.11.51
codio@soundmatrix-clifflemon:~/workspace$ python3 Overflow.py
Traceback (most recent call last):
  File "Overflow.py", line 3, in <module>
    buffer[i]=7
IndexError: list assignment index out of range
codio@soundmatrix-clifflemon:~/workspace$
```

- **What is the result?**

  The compiler throws IndexError with “list assignment index out of range” message.

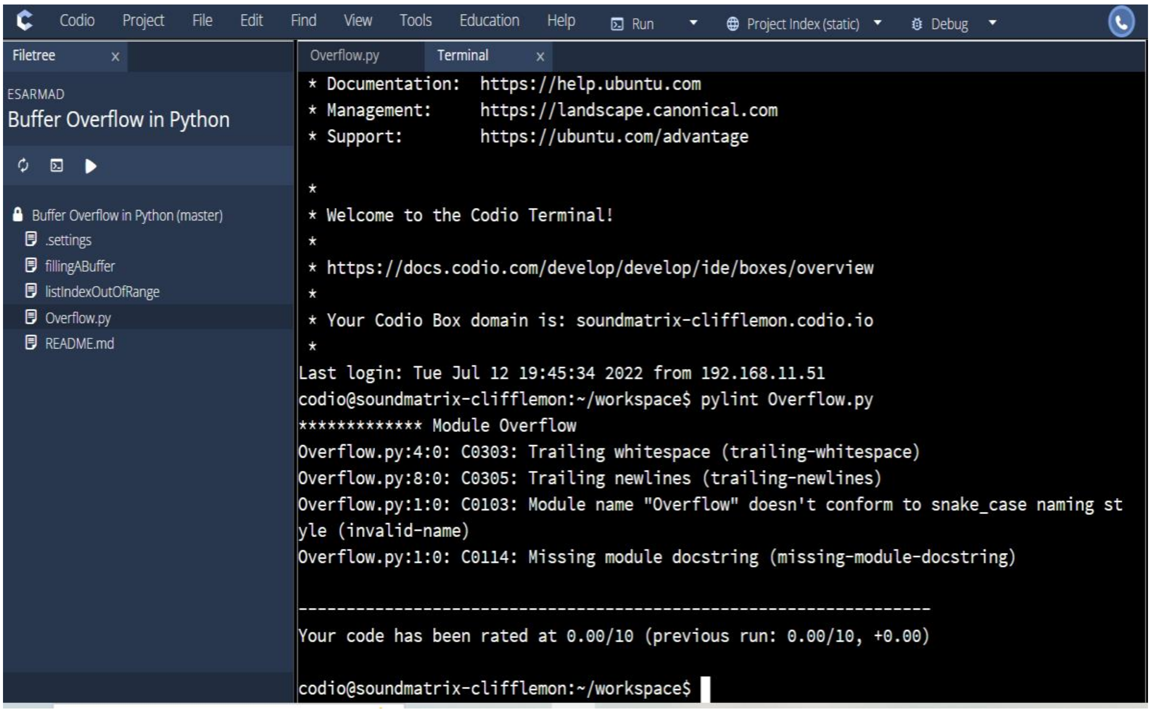
- **Read about Pylint at http://pylint.pycqa.org/en/latest/tutorial.html**
- **Install pylint using the following commands:**

  pip install pylint (in the command shell/ interpreter)

- **Run pylint on one of your files and evaluate the output:**

  pylint your_file e.g.: pylint Overflow.py

- **What is the result? Does this tell you how to fix the error above?**



```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 *
 * Welcome to the Codio Terminal!
 *
 * https://docs.codio.com/develop/develop/ide/boxes/overview
 *
 * Your Codio Box domain is: soundmatrix-clifflemon.codio.io
 *
Last login: Tue Jul 12 19:45:34 2022 from 192.168.11.51
codio@soundmatrix-clifflemon:~/workspace$ pylint Overflow.py
************* Module Overflow
Overflow.py:4:0: C0303: Trailing whitespace (trailing-whitespace)
Overflow.py:8:0: C0305: Trailing newlines (trailing-newlines)
Overflow.py:1:0: C0103: Module name "Overflow" doesn't conform to snake_case naming st
yle (invalid-name)
Overflow.py:1:0: C0114: Missing module docstring (missing-module-docstring)

------------------------------------------------------------------
Your code has been rated at 0.00/10 (previous run: 0.00/10, +0.00)

codio@soundmatrix-clifflemon:~/workspace$
```

(To understand the errors, we use the following command in command line:

Pylint –help-msg=numberOfTheError)

Overflow.py:1:0: *C0114*: Missing module docstring (missing-module-docstring):

It means there is no docstring in our code which violates the convention C0114.

Overflow.py:1:0: *C0103*: Module name "Overflow" doesn't conform to snake_case naming style (invalid-name):

This message belongs to the basic checker which means the name does not conform to naming rules associated to its type (constant, variable, class, …)

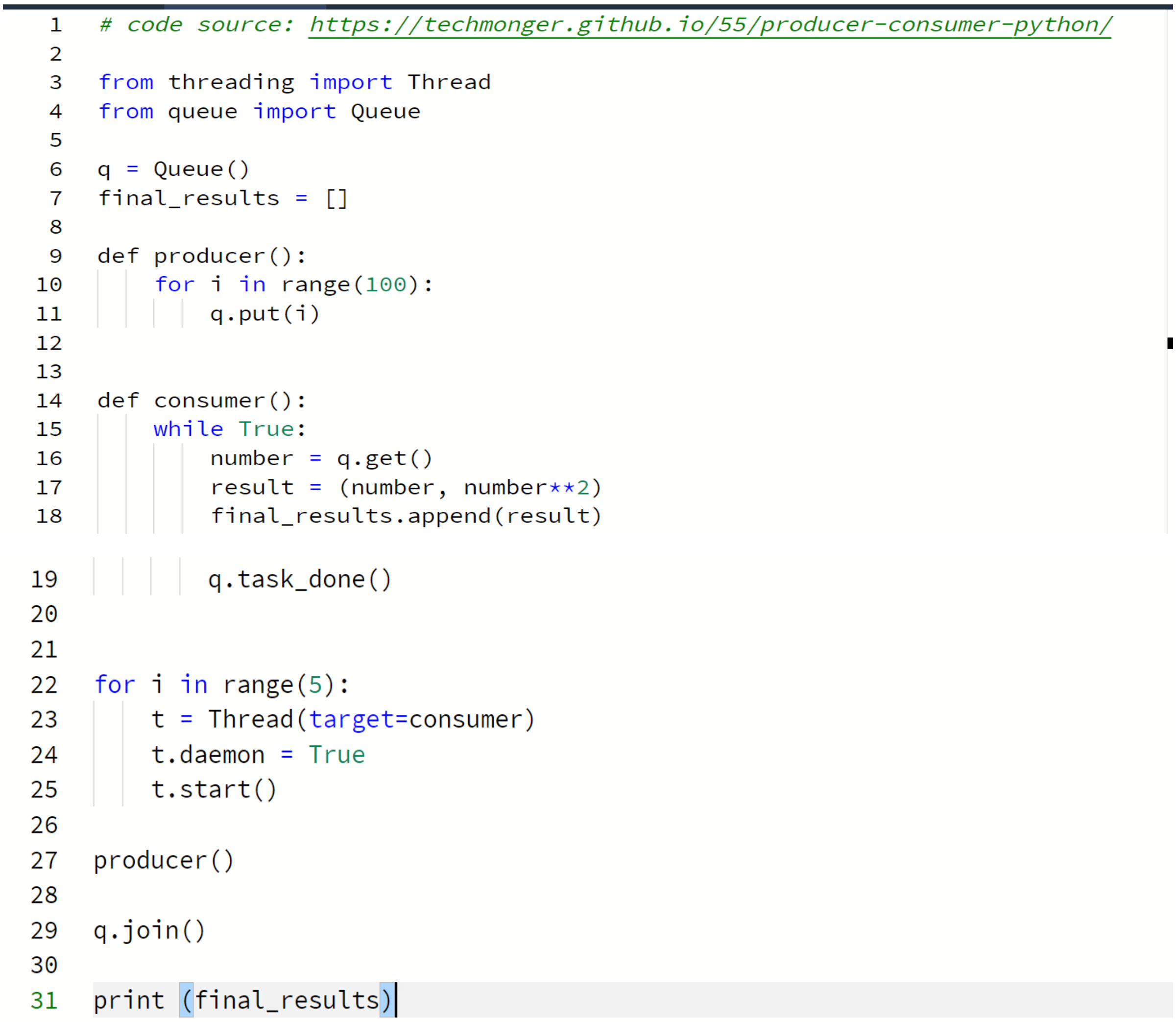
Overflow.py:8:0: *C0305*: Trailing newlines (trailing-newlines):

This message belongs to the format checker which means there are trailing blank lines in a file.

Overflow.py:4:0: *C0303*: Trailing whitespace (trailing-whitespace):

This message belongs to the format checker which means there is whitespace between the end of the line and the new line.

The Producer-Consumer Mechanism (Bounded Buffer Problem)



```python
# code source: https://techmonger.github.io/55/producer-consumer-python/

from threading import Thread
from queue import Queue

q = Queue()
final_results = []

def producer():
    for i in range(100):
        q.put(i)


def consumer():
    while True:
        number = q.get()
        result = (number, number**2)
        final_results.append(result)

        q.task_done()


for i in range(5):
    t = Thread(target=consumer)
    t.daemon = True
    t.start()

producer()

q.join()

print (final_results)
```

1. How is the queue data structure used to achieve the purpose of the code?

Queues are beneficial when resources are shared, and when the speed of the consumers and producers in using the resources is not the same.

2. What is the purpose of q.put(I)?

put() is a method of Queue class which adds items to an instance of Queue class.

3. What is achieved by q.get()?

get() is a method of Queue class which removes items from an instance of Queue class.

4. What functionality is provided by q.join()?

join() is a method of Queue class which blocks the buffer when it is being used by a thread.

5. Extend this producer-consumer code to make the producer-consumer scenario available in a secure way. What technique(s) would be appropriate to apply?

We should adopt techniques that prevent race condition in which threads access the buffer simultaneously. We can use Condition:

```
Cond = threading.Condition()

With cond:

    While some_condition_not_satisfied():

            Cond.wait()
```