Python Assignment 001

- Refer the python code below:

```
#!/usr/bin/python3 -tt

def main():
        var1 = 'Hello'
        var2 = 'World'

if __name__ == '__main__':
        main()
```

- Modify the python code to output the values of the variables as following:

```
shuhari@shuhari:~$ ./p1.py
The value of var1 is: Hello
The value of var2 is: World
shuhari@shuhari:~$
```

Python Assignment 002

- Refer the python code below:

```
#!/usr/bin/python3 -tt

def main():
        first_name = 'Tinku'
        last_name  = 'Pinky'

if __name__ == '__main__':
        main()
```

- Modify the python code to output the following:

```
shuhari@shuhari:~$ ./p1.py
Full name is: Tinku Pinky
shuhari@shuhari:~$
```

Python Assignment 003

- Write a python program that will accept a single command line argument, and print the length of the argument.
- Expected output:

```
shuhari@shuhari:~$ ./p2.py abcd
The length of the first argument passed is: 4
shuhari@shuhari:~$
```

- Assumption: We will assume that the user will enter one or more command line arguments.

Python Assignment 004

Write a python program that will print the following:

- Program / Script Name that was executed.
- Total Number of arguments passed to the program

Expected output:

```
shuhari@shuhari:~$ ./p1.py a b c
The Name of the Program / Script is: ./p1.py
Total Number of Arguments Passed: 4
```

Python Assignment 005

- Write a python program that will accept command line arguments.
- The python program should print the first command line argument and the length of the argument.
- Expected output:

```
shuhari@shuhari:~$ ./p3.py abcd
The first argument passed by the user is 'abcd' and the length is: 4
```

Python Assignment 006

- Refer the python code below:

```
#!/usr/bin/python3

def main():
    var1 = 200
    var2 = 100

if __name__ == '__main__':
    main()
```

Modify the program to contain 4 basic arithmatic functions which will do the operations as mentioned below

- my_add()
  - Takes two arguments
  - Perform addition and return the result.
  - Example: var1 + var2

- my_sub()
  - Takes two arguments
  - Perform substraction and return the result.
  - Example: var1 - var2

- my_div()
  - Takes two arguments
  - Perform division and return the result.
  - Example: var1 / var2

- my_mul()
  - Takes two arguments
  - Perform multiplication and return the result.
  - Example: var1 X var2

Expected output:

```
root@ctf:~/py# ./p1.py
200 + 100 is 300
200 - 100 is 100
200 / 100 is 2.0
200 X 100 is 20000
```

Python Assignment 007

- Write a python program that will check if the user has entered two command line arguments.
- If the user has not entered two command line arguments, print the following message and exit with status code 20.

```
Invalid number of arguments
```

- If the user has entered two command line argument, print the following message and exit.

```
Correct number of arguments passed.
```

Program Output

```
root@ctf:~/py# ./p1.py
Invalid number of arguments
root@ctf:~/py# echo $?
20
root@ctf:~/py# ./p1.py one
Invalid number of arguments
root@ctf:~/py# echo $?
20
root@ctf:~/py# ./p1.py one two
Correct number of arguments passed.
root@ctf:~/py# echo $?
0
root@ctf:~/py# ./p1.py one two three
Invalid number of arguments
root@ctf:~/py# echo $?
20
```

Python Assignment 008

- Write a python program that will prompt the user to enter a number.
- The program should check if the users has entered a value. If not, display an appropriate warning and exit the program.
- If the user has enetered a value, check if it is an integer or not.
- If integer, print the message:

```
The number entered by the user is: 10
```

- If the value is not an integer, print an appropriate message and exit.

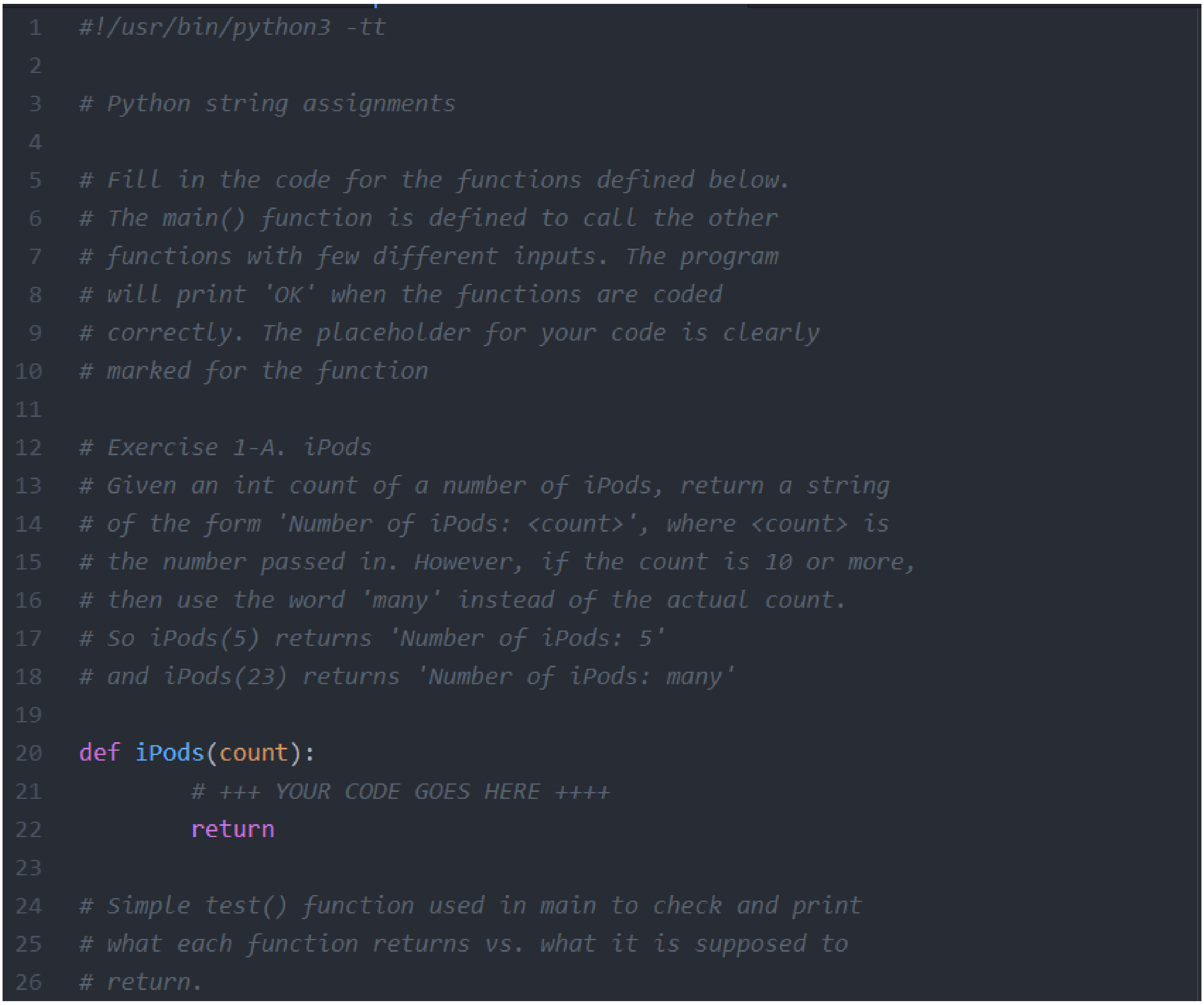
```
shuhari@shuhari:~$ ./p5.py
Enter an integer value:
You are smart. You just pressed the "Enter Key!!!"
shuhari@shuhari:~$ ./p5.py
Enter an integer value: abc
Not a valid integer
shuhari@shuhari:~$ ./p5.py
Enter an integer value: ^%$
Not a valid integer
shuhari@shuhari:~$ ./p5.py
Enter an integer value: 8
The number entered by the user is: 8
```

Python Assignment 009

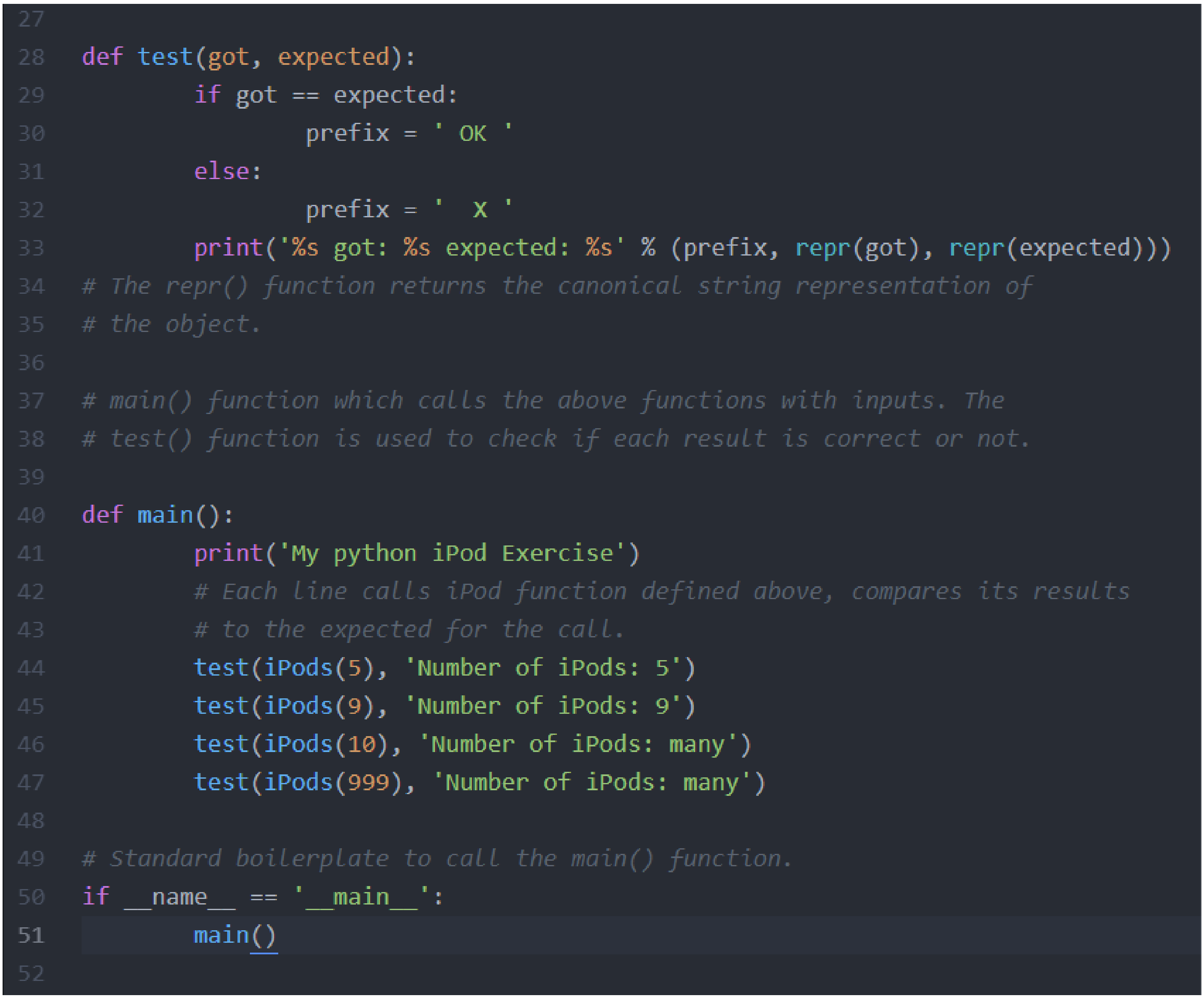
- Write a python program that will prompt the user to enter his / her First Name and Last Name.
- Display the Full Name in French captial letter convention.
- For Example, if the user entered First Name as 'tinku' and Last Name as 'Pinky', display:

```
Your Full Name is: Tinku PINKY.
```

Python Assignment 101



```
#!/usr/bin/python3 -tt

# Python string assignments

# Fill in the code for the functions defined below.
# The main() function is defined to call the other
# functions with few different inputs. The program
# will print 'OK' when the functions are coded
# correctly. The placeholder for your code is clearly
# marked for the function

# Exercise 1-A. iPods
# Given an int count of a number of iPods, return a string
# of the form 'Number of iPods: <count>', where <count> is
# the number passed in. However, if the count is 10 or more,
# then use the word 'many' instead of the actual count.
# So iPods(5) returns 'Number of iPods: 5'
# and iPods(23) returns 'Number of iPods: many'

def iPods(count):
        # +++ YOUR CODE GOES HERE ++++
        return

# Simple test() function used in main to check and print
# what each function returns vs. what it is supposed to
# return.
```



```python

def test(got, expected):
        if got == expected:
                prefix = ' OK '
        else:
                prefix = '  X '
        print('%s got: %s expected: %s' % (prefix, repr(got), repr(expected)))
# The repr() function returns the canonical string representation of
# the object.

# main() function which calls the above functions with inputs. The
# test() function is used to check if each result is correct or not.

def main():
        print('My python iPod Exercise')
        # Each line calls iPod function defined above, compares its results
        # to the expected for the call.
        test(iPods(5), 'Number of iPods: 5')
        test(iPods(9), 'Number of iPods: 9')
        test(iPods(10), 'Number of iPods: many')
        test(iPods(999), 'Number of iPods: many')

# Standard boilerplate to call the main() function.
if __name__ == '__main__':
        main()

```

```
root@ctf:~/pylabs# 
```

```
root@ctf:~/pylabs# 
```