

CSCI 107, Third Practicum – Monday, December 13, 2021

Submit your solutions in a file named *YourFirstName-YourLastName.py* to the CSCI 107 **Practicum 3 Dropbox** no later than 3:50 p.m.

Question One. 25 points. Supply the missing function such that the output below is produced exactly when the code below is run. The missing function is passed a string (e.g. "Hi") and a non-negative integer (e.g. 5). The function should make the appropriate number of copies of each character in the string and pass back that result. For example, if the string is "Hi" and the integer is 5, the function would return "HHHHHiiii".

```
message = "Have a great break!"
for number in range(4):
    print("Result:", duplicate(message, number))
```

```
===== RESTART: C:\Main Directory\Classes\CSCI 107\p3.py =
Result:
Result: Have a great break!
Result: HHaaavvee aa ggrreeaatt bbrreeaakk!!
Result: HHHaaavvvee aaa gggrreeeaattt bbbrrreeeaakkk!!!
```

Question Two. 25 points. Supply the missing function such that the output below is produced exactly when the code below is run. The missing function is passed a string (e.g. "CSCI 107"). The function must use **recursion** to determine the length of the string. Do not use the built-in len function! For example, if the string is "CSCI 107", the function would return 8.

```
print("Test 1:", length_recursive(""))
print("Test 2:", length_recursive("Recursion"))
print("Test 3:", length_recursive("Recursion is fun!"))
```

```
===== RESTART: C:\Main Direct
Test 1: 0
Test 2: 9
Test 3: 17
```

Question Three. 25 points. Supply the missing function such that the output below might be produced when the code below is run. The missing function has two integer parameters and the first parameter will be \leq to the second parameter. The missing function should generate and print random integers that are \geq the first parameter and \leq the second parameter until a random integer that equals the second parameter is generated. At that point, the function should return how many random integers were generated.

```
def main(low, high):
    attempts = generate(low, high)
    print("Low:", low, ", High:", high, ", Attempts:", attempts, "\n")
```

main(10, 10)
main(5, 15)
main(66, 77)

```
===== RESTART: C:\Main Directory\C1
Number generated: 10
Low: 10 , High: 10 , Attempts: 1

Number generated: 15
Low: 5 , High: 15 , Attempts: 1

Number generated: 72
Number generated: 75
Number generated: 72
Number generated: 71
Number generated: 75
Number generated: 69
Number generated: 77
Low: 66 , High: 77 , Attempts: 7
```

Question Four. 25 points. Supply the missing function such that the output below might be produced when the code below is run. The missing function is passed three arguments: the first one is a non-negative integer (e.g. 5) that represents how many rows should appear in the output. The second one is a non-negative integer (e.g. 2) that represents how many columns should appear in the output. The third one is a non-empty string (e.g. "12345") that represents the symbol alphabet. When the function is called, a random symbol from the symbol alphabet should be printed at each coordinate.

matrix(3, 7, "*")
matrix(5, 2, "12345")
matrix(1, 10, "!@#%\$^&*()")

```
===== RESTART: C:\Main Directory\Clas
*****
*****
*****

55
23
22
25
53

&@@#%$^&&
```