

CIS 122 Summer 2015
Project 4: Python Repetition

Due: Thursday 30th July 2015 5:00 PM

Goals:

By the end of this assignment you should be able to:

1. Use while and for loops.
2. Write user input-driven programs.

Getting started: This homework uses the editor window (open Idle, then go to the menu and choose File>New File (may say “New Window” in some versions), a shortcut is ctrl+N on Windows, probably command+N on Macs.

Add comments to indicate the start of a problem using the # sign.

Example:

```
#Problem 1
```

The # sign tells Python to ignore everything after it.

Problem 1: project4a.py

20 points

Taxman:

10 points

Using the Tax Slab from the previous project, extend the tax program to run as long as the user does not enter ‘quit’ as their name. In this program, you must keep track of the number of entries and calculate and print the following:

- Average Tax Collected
- Total Tax Collected
- Number of persons

Odds and Evens:

10 points

Write a function, Odds_or_Evens(), to match the following docstring. Remember to include the docstring with your code.

```
"""
```

```
(int, bool) -> int
```

Odds_or_Evens() takes two arguments, a max number to count to and a Boolean. If the Boolean is true, then the function will print all even numbers up to the max number. If the Boolean is false, then Odds_or_Evens() prints all odd numbers up to the max number.

```
>>> Odds_or_Evens(10, False)
```

```
1 3 5 7 9
```

```
>>> Odds_or_Evens(7, True)
```

```
0 2 4 6
```

```
"""
```

XC: +5 points, make the above function user_input driven while re-using our Odds_or_Evens() function.

Usage:

```
>>>user_odds_or_evens()
```

Menu

```
1.Count the evens
2.Count the Odds
Choice? 1
Till what number? 7
0 2 4 6
Continue(y/n)? n
Bye!
>>>
```

Problem 2: project4b.py

20 points

Two numbers

10 points

Define a function, two_numbers, with one parameter, astring, which returns True if astring has at least two numbers, and otherwise returns False.

Special Character Checker

10 points

Define a function, any_special_char with one parameter, astring, which returns True if astring contains any of the special characters, '!', '@', '#', '\$', '%', '^', '&', and otherwise returns False.(Hint: define a string variable which has the value of a string containing all of the special characters.)

XC: +10 points Convert Both the above programs into looping programs that prompts the user if they want to re-run the program with a different string.

Problem 3: project 4c.py

Name Game

10 points

Write a function, Name_Game(), to match the following docstring. Remember to include the docstring with your code.

```
"""
```

```
() -> None
```

Name_Game() takes no arguments but asks the user to input their name. It then prints out the lines from the name game. Those lines are as follows:

```
<name>,<name> , bo-b<name except first letter>,
Banana-fana fo-f<name except first letter>
Mee-Mi-mo-m<name except first letter>
<name> !
```

examples (doctest won't work since requires user input):

```
Name_Game()
What is your name?
Katie
Katie, Katie, bo-batie,
```

Banana-fana fo-fatie,
Mee-Mi-mo-matie,
Katie!
Name_Game()
What is your name?
Sharon
Sharon, Sharon, bo-bharon,
Banana-fana fo-fharon,
Mee-Mi-mo-mharon
Sharon!
""

Hint- the name game is actually more complicated than the above but just worry about matching what I've shown.

XC: +5 points, make it so that prompts the user to continue after every iteration and only if the user enters 'y' or 'yes' call Name_Gamme()