****

In the pre-class videos, you learned about if statements (if-else, if-elif, if-elif-else) and while loops. We'll explore these more in this lab.

1   INPUT VALIDATION

1. While loops are often used for input validation. Say we want a user to enter the number 1 or 2. Analyze the blank below. What code could you place in the blank so that the user is continually asked to enter a 1 or 2 until the user actually does so?

```
01.   #Input Validation Program
02.
03.   x = int(input("Enter 1 or 2 please: "))
04.
05.   while   x != 1 and x != 2          :  #check to see if x is neither 1 nor 2.
06.       x = int(input("Enter 1 or 2 please: ")) #Re-ask for a 1 or 2.
07.
08.   print("\nFINALLY!")
```

2. Add or Copy/paste the contents of the Lab3A.py file into Spyder. Uncomment the task 1 area and include your guess from above. If it works, you should see output like that shown at the right.

```
Enter 1 or 2 please: 5
Enter 1 or 2 please: 3
Enter 1 or 2 please: 4
Enter 1 or 2 please: 6
Enter 1 or 2 please: 1
FINALLY!
```

3. Sometimes we want to know how many times a loop is run, that is, how many iterations of the loop occur. We often do this by using a counter that increments each time a loop occurs. Between lines 3 and 4 above, initialize a variable called counter to 1.

```
counter = 1
```

4. Now, add a line of code in the while loop/suite so that counter increments by 1 each time the loop occurs, thus counting how many times the while loop occurs. Update the final print statement to print this information out, like in the example below.

```
Enter 1 or 2 please: 4

Enter 1 or 2 please: 4

Enter 1 or 2 please: 4

Enter 1 or 2 please: 3

Enter 1 or 2 please: 2

FINALLY! It took you 4 times - geesh!
```

5. In the same file, below the code above, add new code that does the following: Keep asking the user to input an even number until he or she does so, and then print out how many times it took, like below.

```
Enter an even number please: 5
I SAID INPUT AN EVEN NUMBER: 3
I SAID INPUT AN EVEN NUMBER: 7
I SAID INPUT AN EVEN NUMBER: 9
I SAID INPUT AN EVEN NUMBER: 10

FINALLY! It took you 5 times to remember what an even number is!
```

1. In your homework due this week, you had to make the left table below print out. But what if we wanted to be able to print the table out longer like to the right? And what if we want the user to tell us how long this table should be? Well, really, we just need to find the pattern and let a loop do the work for us.

```
                                    Enter an how many rows you'd like to see: 10
                                    number   square  cube
number  square  cube                0        0       0
0       0       0                    1        1       1
1       1       1                    2        4       8
2       4       8                    3        9       27
3       9       27                   4        16      64
4       16      64                   5        25      125
5       25      125                  6        36      216
                                     7        49      343
                                     8        64      512
                                     9        81      729
```

2. Read the code below and answer the question above it. Fill in the blanks of the code so that when the user enters 10, 10 rows are printed out (just like the above right table). Then uncomment out this area in the code file and try your guess.

Question: Why does the table end at number 9 in the upper right table if the table has 10 rows?

It is 0 indexed
_____

_____

```
01.   #Ask the user for how many rows.
02.   numRows = int(input("Enter an how many rows you'd like to see: "))
03.
04.   #Set the counter to a starting value.
05.   counter = ____0_____
06.
07.   #print the header
08.   print("number\tsquare\tcube")
09.
10.   #Use a loop to print out the rest of the rows.
11.   while (counter ___< numRows_____):
12.       print(____counter, '\t', counter ** 2, '\t' counter_**3
13.       counter = counter += 1_____
```

3   USING COUNTERS TO FIND THE PATTERNS IN YOUR LAST HOMEWORK!

1. Recall that you wrote homework to separate a given 5 digit number into 5 distinct digits. We'll update that problem to ask the user to input a number of ANY length and then we'll use a loop to separate the digits.

Say the user entered 11334. For most of you, the algorithm you all used was:

STEP 1: Get the rightmost digit (mod 10).   Example: 11334 % 10 gives 4
STEP 2: Remove the rightmost digit (do an integer division using "//").   Example: 11334 % 10 gives 1133

2. Because a pattern was used, a loop will work nicely to do this. However, with what we've learned so far, the problem is that we'd like to get the LEFTMOST character first, print it, and then remove it. We can still use "//" and "%" to do this if we just use powers of 10. Fill out the table below to get a feel for this.

x	integer division (truncation)	Remainder
11334	x // 10000 = 1	x % 10000 = 1334
1334	x // 1000 = 1	x % 1000 = 334
334	x // 100 = 3	x % 1000 = 34
34	x // 10 = 3	x % 10 = 4
4	x // 1 = 4	x % 1 = 4

3. Based on this discussion, fill in the blanks and follow the comments to solve this problem with a while loop.

```
01.   #Get input
02.   x = input("Input an integer: ")
03.
04.   #The len(string) function returns the length of a string
05.   #For example, len("Python") evaulated to 6
06.   n = len(x)
07.
08.   #A line of code goes here.  Hint:  What type of variable do you want x to be?
09.          x = int(x)
10.   #set your counter
11.   counter = _____x = 1_____
12.
13.   while _counter < n_____:
14.        #Do the expected truncation/remainder logic to get the leftmost digit.
15.        #print the leftmost digit
16.        #Update the counter as needed
```

4   PRACTICE WITH IF STATEMENTS: AGE CLASSIFICATION

You learned that the basic structure of the if statements are as follows.

```
if condition:      #case 1
    do something.

elif condition:    #case 2
    do something.

...etc...

elif condition:    #case n
    do something.

else:              #catch-all for all other cases
    do something for every other case possible.
```

With this in mind, write a program that asks the user to enter a person's age. The program should display a message indicating whether the person is an infant, a child, a teenager, or an adult. Following are the guidelines:

• If the person is 1 year old or less, he or she is an infant.
• If the person is older than 1 year, but younger than 13 years, he or she is a child.
• If the person is at least 13 years old, but less than 20 years old, he or she is a teenager.
• If the person is at least 20 years old, he or she is an adult.

TO GET CREDIT FOR THIS LAB, UPLOAD THESE 2 DOCUMENTS TO THE SUBMISSION AREA.

- LabCh3A.pdf (should have your guess work)
- LabCh3A.py (make sure you update the header with your name)