**ENGR 102 Sect 508 Lab 12b**

*Attention!!*

*Individual submission. one submission per person.*

*Submit* *your Py-files together with your word/pdf file with screenshots of your tests outputs. Include any derivations, comments and supplemental notes in your word/pdf files.*

*No pictures by the phone – it is impossible to read. You will be allowed to resubmit and reupload HW as many times as you want to within the due date/time, only last submission will be graded. No late submissions. For submission you may use this file as template: rename file including your name. Do not forget to put your name inside of this file as well. If it is a team work use Team Header, include the team number and all team members.*

**Activity #2: Creating tally marks – to be done individually**

You are to create a program that uses the turtle graphics routines to draw tally marks representing numbers. Briefly, a user should be asked for a number, and then tally marks should be drawn equivalent to the number.

• Tally marks are used to aid counting. As numbers are counted, each number encountered is represented by a short vertical line, except that every fifth number will instead be represented by a diagonal line passing through the previous four vertical lines.

• Your program should be able to draw tallies for numbers up to at least 100.

**due 11/19/18**

• Each group of 5 tallies should be clearly separated from other groups.

• When you have to draw tallies, each “row” of the drawing should be able to hold at least 10 (i.e.

2 sets of 5) and at most 25 (i.e. 5 sets of 5) tally marks. You may choose how many a row should hold (anywhere from 2 to 5 sets of 5), but should make sure that a window easily displays up to 100 tallies.

• You may find the Python documentation of the turtle module helpful to refer to: https://docs.python.org/3.3/library/turtle.html

You may use a top-down, a bottom-up, or some hybrid (like in the team activity) approach to designing your program. But, you should go through a design process, first.

a) Before writing any code, you should go through a design process. Try to do so carefully – either follow a top-down approach, a full bottom-up approach, or some combination. But, in any case, you should come up with a list of functions you plan to implement, each with its own clear purpose.

a. Create a document that outlines all the functions you will create, including a purpose for each, stating what the function does, and any parameters it has. The purpose for each function can become its docstring

b) After you have written a description of your program, including the planned functions, go ahead and write code for each function. Be sure to create a docstring for each function. Turn in your program and the design document you created beforehand, zipped into one file.

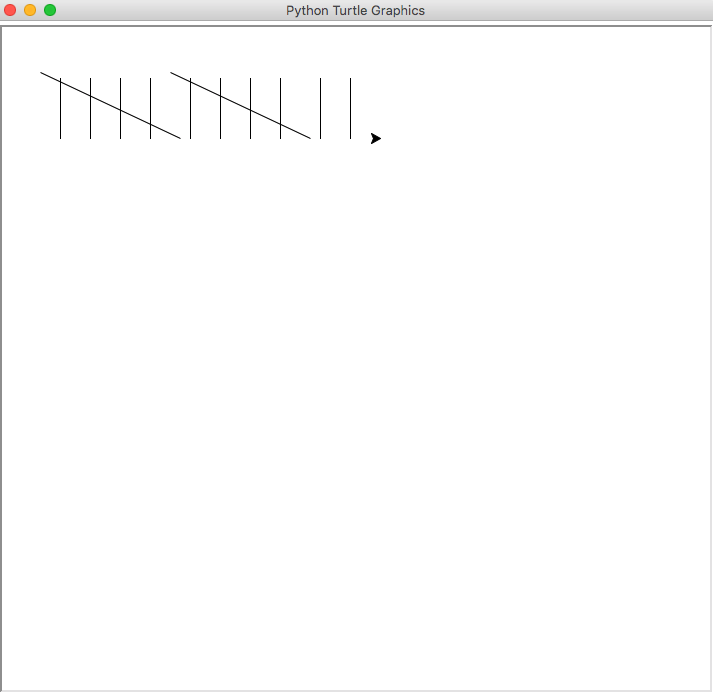
Challenge: See if you can create a program so that the lines are drawn with some irregularities, so that each is not perfectly vertical, or not perfectly straight, etc.

Code:

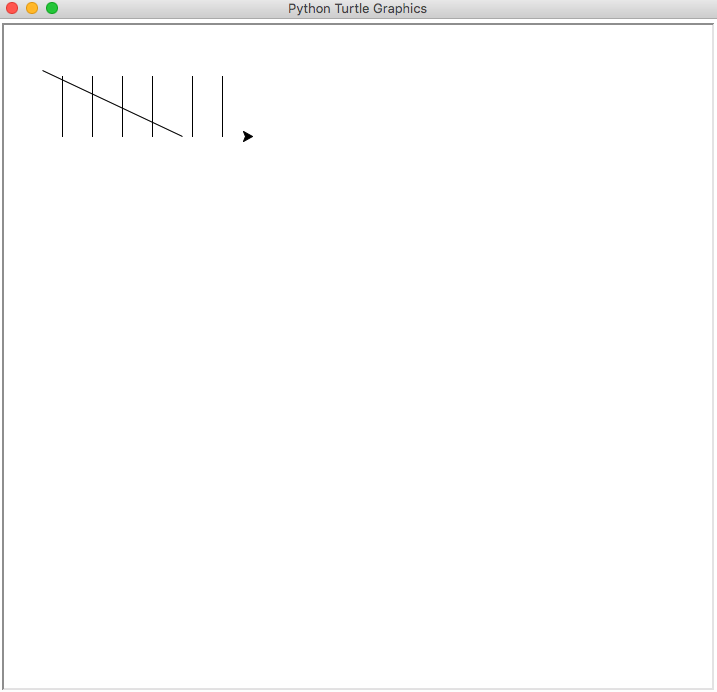
*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 12B  
# Date: 16-11-2018***from** math **import \*  
import** turtle  
  
t **=** turtle.Turtle()  
num **=** int(input(**'Enter a positive number: '**))  
  
  
**def tally**()**:** t.left(90)  
 t.forward(60)  
 t.up()  
 t.right(90)  
 t.forward(30)  
 t.right(90)  
 t.forward(60)  
 t.left(90)  
 t.down()  
  
  
**def five\_tally**()**:  
 for** i **in** range (4)**:** tally()  
 t.left(155)  
 t.forward(155)  
 t.up()  
 t.left(205)  
 t.right(90)  
 t.forward(65)  
 t.left(90)  
 t.forward(20)  
 t.down()  
  
  
**def draw**(*num*)**:** z**= -**300  
 a **=** *num* **%** 5  
 b **=** *num* **//** 5  
 y **=** 225  
  
 **for** i **in** range(b)**:** t.up()  
 t.setposition(z, y)  
 t.down()  
 t.speed(10)  
 five\_tally()  
 z **+=** 130  
 **if** t.position()[0] **>** 200**:** z**= -**300  
 y **-=** 100  
 t.up()  
 t.setposition(z,y)  
 t.down()  
 **for** i **in** range(a)**:** t.up()  
 t.setposition(z,y)  
 t.down()  
 t.speed(10)  
 tally()  
 z **+=** 30  
 **if** t.position()[0] **>** 200**:** x **= -**170  
 y **-=** 100  
 t.up()  
 t.setposition(x,y)  
 t.down()  
  
  
draw(num)  
input()

Outputs:

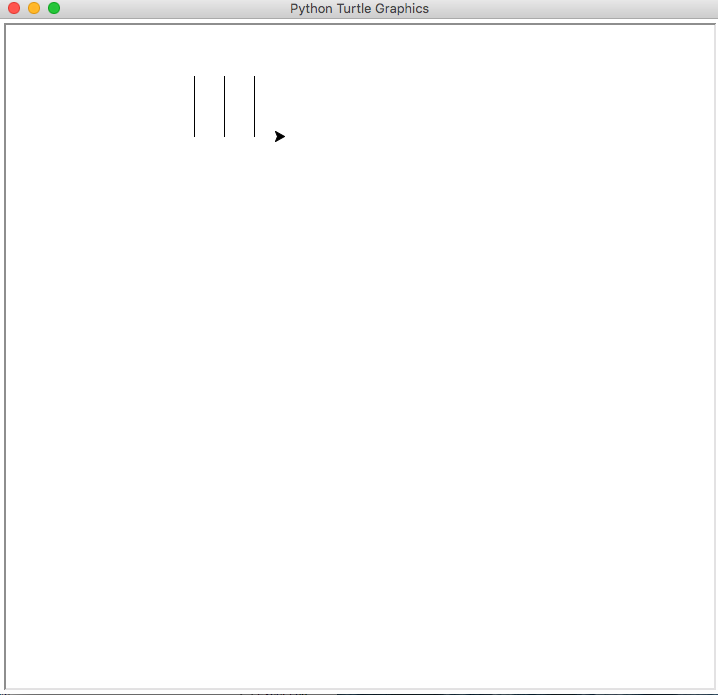
Test # 12:



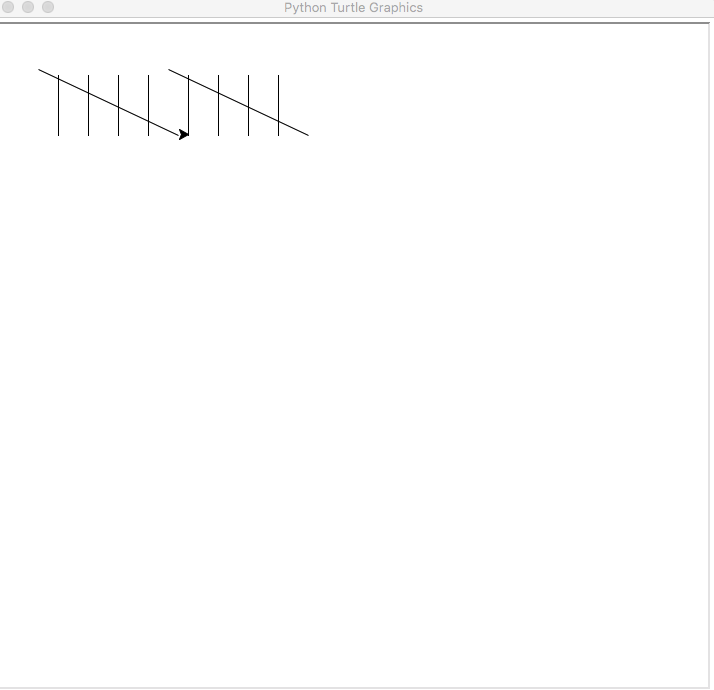
Test # 7:



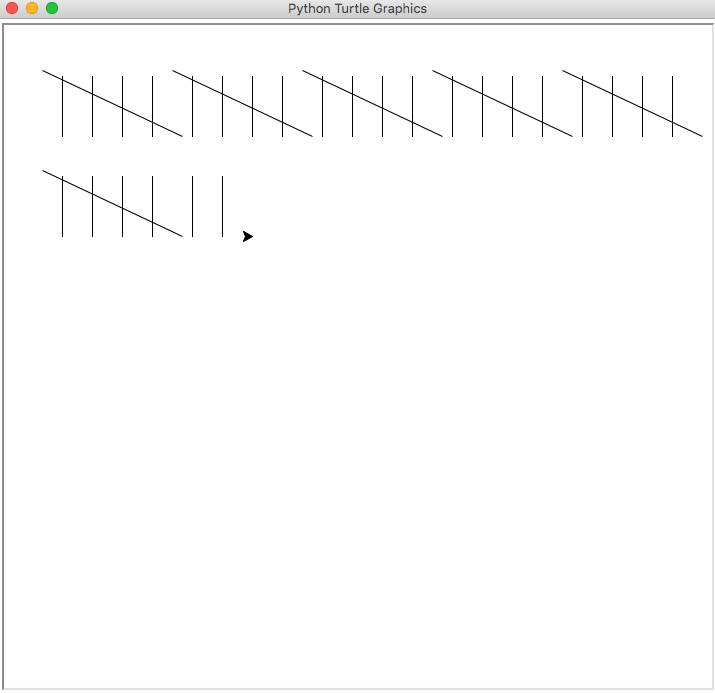
Test # 3



Test # 10



Test # 32



Test # 45:

