CIS 122 Winter 2016 Project 3
Due Monday Jan. 25, Midnight

Submit each program as a separate file.

25 points

P3_if_numbers.py **5 points**

1 point
1) Create two variables, **count** and **limit**
Assign a **5** to **count,** assign **3** to **limit**.
If **count** is **less than limit**, print a message showing count and announcing that count is less than the limit
Else, print a message showing count and announcing that count is NOT less than the limit.

2 points
2) Create a list of amounts in inventory like this
`amounts = [5, 16, 44, 31, 107, 48, 22, 999, 8]`
Print an inventory report on each item
Set **message** to "**OK**"
For items less than `low_limit` of **10**
Set **message** to "**Too few items**"
For amounts above `high_limit` of **99**
Set **message** to "**Too many items**"
For amounts equal to `missing` –it's **999**
Set **message** to
"**Unknown number of items - recheck inventory**"
Print amount and message

2 points
3) Create a list of prices at the Cohnah Market like this
`prices_list = [4.95, 9.90, 12.44, 1.99, 27.95, 5.00, 11.05, 144.23, 20.00]`
Print a report on each price
Set **message** to ""
For price **less than or equal** to `bargain` of **5.00**
Set **message** to "**Sale price!**"
For price **at or above** `quality` of **20.00**
Set **message** to "**Exclusive offering**"
Print price and message

```
--
Count 5 NOT less than limit 3

Inventory
5 Too few items
16 OK
44 OK
31 OK
107 Too many items
48 OK
22 OK
999 Unknown number of items - recheck inventory
8 Too few items


Cohnah market's prices
4.95 Sale price!
9.9
12.44
1.99 Sale price!
27.95 Exclusive offering
5.0 Sale price!
11.05
144.23 Exclusive offering
20.0 Exclusive offering
```

P3_if_text.py **3 points**

Create several variables with these values
word1 = 'Book'
word2 = 'art'
word3 = 'artist'
word4 = 'book'

1 point
Compare word2 with word3.
if word2 comes before word3, print word2 "comes before" word3
else print word2 "comes after" word3

1 point
Compare word1 with word2.
if word1 comes before word2, print word1 "comes before" word2
else print word1 "comes after" word2

1 point
Create this variable and this list
name = 'Wenlan'
chosen = ['Bob', 'Jan', 'Wenlan', 'Ashley']
check whether name in chosen list
if name in chosen, print(name, "is in list")
if not, print(name, "is NOT in list")

P3_elif.py **10 points**

The Python if, elif, elif, else statements allow assigning a data item to one and only category. **elif** means "else if".

Example

```
# P3_elif_demo.py

height = 54

if height >= 78:  # 6 feet 6 inches
    category = "very tall"

elif height >= 72: # 6 feet
    category = "tall"

elif height >= 62: # 5 feet 2 inches
    category = "medium"

elif height >= 54: # 4 feet 6 inches
    category = "short"

else:
    category = "very short"

print(height, "inches", category, "group")
```

After getting a value such as "very short" to category, Python jumps past all remaining elif and else; it then prints height and category.

```
54 inches short group
```

For this part of the assignment, use **if elif elif else** ...

You will ask for a name, or "Quit" then for the number of points earned on a test. Using this table below, assign a letter grade, print the name, points and grade, then repeat.

```
# grade points
grade_A =  57
grade_B =  43
grade_C =  33
grade_D =  25
```

2 points
Use a **while** loop to repeatedly ask for a name or "Quit".

2 points
Ask for number of **points** earned.

5 points
Use an **if elif** structure to assign a value such as "A" or "B" based on the number of points.

1 point
Ask again for name or "Quit".

Results may look like this (notice a test for each grade):

```
First name (or Quit to end): Morgan
Points on test: 57
Morgan 57 A
First name (or Quit to end): Norma
Points on test: 43
Norma 43 B
First name (or Quit to end): Oscar
Points on test: 33
Oscar 33 C
First name (or Quit to end): Pam
Points on test: 25
Pam 25 D
First name (or Quit to end): Quentin
Points on test: 24
Quentin 24 F
First name (or Quit to end): Quit

Finished
```

Bonus + 1 Collect names and points in 2 lists, print grades after all data collected.

After collecting the names and points into 2 lists, calculate grades and print name, points and grade for each student.
Your results will look something like this:

```
First name (or Quit to end): Morgan
Points on test: 57
First name (or Quit to end): Norma
Points on test: 43
First name (or Quit to end): Oscar
Points on test: 33
First name (or Quit to end): Pam
Points on test: 25
First name (or Quit to end): Quentin
Points on test: 24
First name (or Quit to end): QUIT

Morgan 57 A
Norma 43 B
Oscar 33 C
Pam 25 D
Quentin 24 F

Finished
```

P3_Spiral.py **7 points**

Never *save a file called "***turtle.py***" – if you do so, turtle graphics will not work on your computer until you change the name to anything else.*

To `fill` *a triangle with a* `blue` *color, do this*

```
#P3_demo_fill.py

import turtle as t

def draw_triangle(size):
    angle = 360 / 3
    for count in range(3):
        t.forward(size)
        t.left(angle)
    return None

length = 120

t.fillcolor('blue')
t.begin_fill()

draw_triangle(length)

t.end_fill()
```

1 points Set up to draw a spiral as before
Set **pencolor** to "**red**"
Set **pensize** to **2.**
Set **speed** to '**fastest**'
Set the variable **size** to **20**
Set the variable **angle** to 360 divided by 4
Set the variable **nudge** to **4.7** (any amount from 3 to 8 could work here)
Set the variable **bump** to **5**

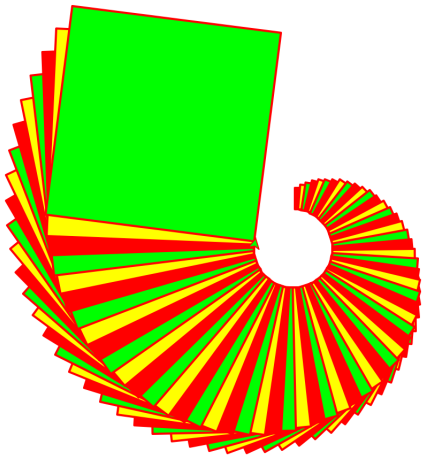
1 point
Repeat the following 60 times

4 points

Draw a square with side length **size**
filled with the next color from a **color_list** of at least 3 colors

1 point
Turn the turtle right **nudge** degrees
Move forward **bump** units
Change **size** to be **4** units larger

Your drawing will look like this — you might have some different color choices, or number of colors, and your spiral might curve to the left, or to the right, but it should show some filled squares looking more or less similar to this:



Bonus + 1 point
Fill the color list by asking a user to name the colors to use.