Goals
By the end of this assignment, you will have
- practice with Python Boolean data type
- practice with Python conditional (if) statements
- practice with Python string methods
- more practice with docstrings and good programming style

(0) OK or Not OK?

Write a function that takes one input parameter, word, of type string and returns True if word is at least 5 characters long, and False otherwise.

Write a function that takes one input parameter, word, of type string and returns True if word contains at least one non-alphanumeric character, and False otherwise. HINT: investigate string method isalnum.

Write a function that takes one input parameter, word, of type string and returns True if word does NOT contain the characters 'E' or 'e', and False otherwise.

(1) Grade Calculator

With one midterm and nearly half of the projects for the term completed, you may be interested in checking your progress in CIS 122 so far. A Python function would be perfect for helping with this tedious work, and the grade_calculator function found here does just that: http://www.cs.uoregon.edu/Classes/14S/cis122/project5star.py.

Start by entering the following function into a new Python file. (Cutting and pasting from a .pdf is not recommended, but you can download this code from the class website or, probably best, type it in yourself.)

def grade_calculator():
    """() -> None

    Prompt user to input a list of grades
    Adjust grades to correct weight
    Prints total grade as percent

    >>> grade_calculator()
Enter project 1 grade: 25
Enter project 2 grade: 38
Enter project 3 grade: 28
Enter project 4 grade: 35
Enter midterm 1 grade: 39
 83 B  """
proj_so_far = 14  #4 projects, 3.5 weighted pts. each
tests_so_far = 20  #1 midterm 20 pts. weighted
ttl_points_possible = proj_so_far + tests_so_far

p1adjust = 1.17  # 30 ttl pts, multiply by 1.17 to get to 35
p2adjust = .875  # 40 ttl pts, multiply by .875 to get to 40
p3adjust = 1.17
p4adjust = .875
last_adjust = 10  # now divide by 10
m1adjust = .4  # 50 ttl pts, multiply by .4 to get to 20

# initialize weighted totals
proj_grade = 0
exam_grade = 0

p1 = int(input('Enter project 1 grade: '))
proj_grade += p1 * p1adjust

p2 = int(input('Enter project 2 grade: '))
proj_grade += p2 * p2adjust

p3 = int(input('Enter project 3 grade: '))
proj_grade += p3 * p3adjust

p4 = int(input('Enter project 4 grade: '))
proj_grade += p4 * p4adjust

proj_grade /= last_adjust

m1 = int(input('Enter midterm 1 grade: '))
exam_grade = m1 * m1adjust

my_ttl = proj_grade + exam_grade
my_score = my_ttl / ttl_points_possible * 100
my_score = round(my_score)
print(my_score, end=' ')

pass  #Your code goes here

return #None
grade_calculator()

We will talk about this code in lab this week. You should also test it to make sure it works and read through it so you understand what is happening on each line.

Edit the function so it also prints the letter grade that corresponds to the percent score, where a percent greater than or equal to 90 is an A, greater than or equal to 80 is a B, greater than or equal to 70 is a C, greater than or equal to 60 is a D, and lower than 60 is F.
(2) Safe Lead?

Implement the "safe lead" algorithm, designed by sportswriter Bill James, which is designed to answer the question, Under what conditions can you safely determine that a lead in a basketball game is insurmountable? [Those of you who were little kids or hang out with them now will be aware that this algorithm does not apply to kids' games!]

The algorithm is as follows:

- Take the number of points one team is ahead
- Subtract three
- Add $\frac{1}{2}$ point if team that is ahead has the ball, subtract $\frac{1}{2}$ point otherwise
- Square the result
- If the result is greater than the number of seconds left, the lead is safe

Write a function, safe_lead, that takes a single parameter, lead, and returns True if the lead is "safe", according to the above algorithm, and returns False otherwise. The function should print a message reporting the result before returning the appropriate value. The user of the function should be prompted to enter whether the team in the lead has the ball and the number of seconds remaining in the game.

(3) Ahead of the Game: Write a function, tax_calc, with one parameter, tincome (taxable income). The function should compute and return the tax for this (unadjusted) income. For the first $9,225 of earned income, the tax rate is 10%. Any income over $9,225 but less than or equal to $37,450 is taxed at 15%. Any income over $37,450 but less than or equal to $90,750 is taxed at 25%. Any income over $90,750 is taxed at (for the purposes of this calculator) at 28%.

Test your code using the follow data:

```python
>>> tax_calc(5250)
525.0
>>> tax_calc(30000)
4038.75
>>> tax_calc(40000)
5793.75
>>> tax_calc(90750)
18481.25
>>> tax_calc(100000)
21071.25
```
(4) Population Density Analysis: Write a function, `density_rpt`, with two parameters, `population` and `land_area`. The function should print the land density (number of people per unit of area). If the land density is greater than 100, the function should print "Densely populated" and "Sparsely populated" otherwise. None value is returned.

Test your code using the following data:

```python
# Oregon
density_rpt(3899000, 98381)
Land density is 40 per square mile.
Area is sparsely populated.

# Washington
density_rpt(6897000, 71300)
Land density is 97 per square mile.
Area is sparsely populated.

# Idaho
density_rpt(1596000, 83570)
Land density is 19 per square mile.
Area is sparsely populated.

# California
density_rpt(38040000, 163695)
Land density is 232 per square mile.
Area is densely populated.
```
(Challenge)

(a) From codingbat.org: Write a function, make_chocolate, with three parameters, small, big, and goal. We want make a package of goal kilos of chocolate, from the available small and big bars of chocolate. The small bars are 1 kilo each and the big bars are 5 kilos each. Big bars cannot be divided; they must be used all at once. Return the number of small bars to use, assuming we always use big bars before small bars. Return -1 if it can't be done.

```python
>>> make_chocolate(4, 1, 9)
4
>>> make_chocolate(4, 1, 10)
-1
>>> make_chocolate(4, 1, 7)
2
>>> make_chocolate(3, 1, 9)
-1
>>> make_chocolate(6, 2, 10)
0
```

Do not use iteration (looping) in your solution. For hints, see the end of this project description.
Grading Rubric

This project will be marked out of 70 points. Each function (3 OK or Not OK, grade_calculator, is_safe_lead, tax_calc, and density_rpt) is worth 10 points: 2 points for the function header, 3 points for the docstring, 2 points for the body of the function, 1 point for the return statement, and 2 points for correct output.

Getting Started
Write your program using an editor window, so you can save your file.

Programming style is important! Remember
- Include a docstring in every function per example in text p. 47 (type contract, function description, examples of function calls)
- Use whitespace between operators and operands
- Use descriptive variable names
- Add appropriate comments.

Examples are given for many but not all of the function calls. Where they are not given, you will need to supply them. The examples help explain the function and also can be used to test the function. You do not need to turn in the example function calls.

Finishing & submitting your work

When you have completed all of the problems, add additional comments to your code to make sure the functions for each problem are clear. Check the list of functions above to make sure that your Python file includes all of them. Do a final Save command to save the code in the editor window as a file with the name cis122project5.py

To submit your project, login to Blackboard. From the menu on the left hand side of the screen, choose "Projects". In the projects folder, choose "Submit Make Good Choices". In Section 2 of the page that is displayed, scroll down to "Attach file" and choose "Browse My Computer". Locate cis122project5.py (the file you just created), and double click on it. The file name will appear on the "Attached Files" list.

At the bottom of Section 2, you will see a "Comments" window. This is where you credit all of the sources of any help you may have received on this assignment, including your partner if you are working in a programming pair. This is also the place to include any feedback you may have about the assignment and/or any remaining questions you may have.

Scroll down to Section 3 and hit the "Submit" button. You may re-submit your project up until the project deadline. Only the final submission will be graded.

Hint #1 for make_chocolate: Use % (modulo, remainder) operator.