PROBLEM 1 (*Simple Function*) Part A

Lab Employee Pay

Your labs supervisor is interested in creating a system for calculating weekly pay based on each employee's classification level and number of hours worked. Undergraduates and graduate students earn 15 per hour. Postdoctoral researchers earn 25 per hour. Staff earn 35 per hour. Undergraduates and graduate students cannot be paid for more than 20 hours in one week. Thus, if they have worked more than 20 hours in a week, then they will only be paid for the first 20 hours. Postdoctoral researchers and stff have no limit on the number of hours that they can be paid for.
Create the function calculatePay that has two parameters: level is a string representing the classification level of each lab member (ugrad is undergraduate, grad is graduate student, post is postdoctoral researcher, and staff is staff member); and hours is a float representing the number of hours worked. The calculatePay function calculates and returns the total pay, based on the information provided above.

call	returns	comment
calculatePay('ugrad', 24.0)	300.0	undergrads only paid for the first 20 hours at 15/hr
calculatePay('staff', 35.5)	1242.5	staff members paid at 35/hour with no hour limitations
calculatePay('post', 10.5)	262.5	postdocs paid at 25/hour

Complete the function calculatePay below.

```
def calculatePay(level, hours):
```

Part B

$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Above is the formula to calculate the distance between two points: (x_1, y_1) and (x_2, y_2)). Implement the function `distance` that takes the x- and y-coordinate for two points and returns the distance between them. Assume that the `math` library has already been imported for you.

```
def distance(x1, y1, x2, y2):
    """
    x1 (int/float) - x-coordinate for first point
    y1 (int/float) - y-coordinate for first point
    x2 (int/float) - x-coordinate for second point
    y2 (int/float) - y-coordinate for second point

    Return the distance between the two points.
    """
```

PROBLEM 2

Part A

Iteration and recursion. In this question we approach a simple problem, that of making a reversed copy of a list (without modifying the original list), in two ways.

(a) Implement the following function according to spec, using **recursion**. Your function must call itself.

```
def reverse(x):
"""Return a copy of the list x, in reverse order."""
```

(b) Implement the following function according to spec, using **iteration**. Your function must be based on a loop. Hint: There are solutions based on counting forwards, counting backwards, and looping over the sequence directly; use whichever you find easiest.

```
def reverse(x):
"""Return a copy of the list x, in reverse order."""
```

Part B

Write a list comprehension that creates a list of all the multiples of 5 greater than 0 and less than 4096,
e.g.,[5,10,15,...,4095].

PROBLEM 3 :

You're hired by Apple to manage the purchases of Beatles tracks. A file of purchases is kept with each title stored with the email addresses of the people who have purchased that track. Three tracks are shown below. The title is the first string on each line, separated from email addresses by a comma. Each email address for a given track is separated by a comma as well

Drive My Car,ola@duke.edu,pjl@msn.com
Norwegian Wood,pjl@msn.com,jf@foo.edu
Nowhere Man,pkp@nbc.com,pjl@msn.com,joa@gmail.com

Write a function bbFan that returns the email address of the biggest Beatles fan – the email address of the person who purchased the most tracks. The function is passed the name of the file storing the data.

```
def bbFan(filename):
  file = open(filename)

  file.close()
```

PROBLEM 4 : Classes and Subclasses

Write a class named Employee that holds the following data about an employee in attributes: name, ID number, department, and job title.

Once you have written the class, write a program that creates three Employee objects to hold the following data:

Name: Ayman Nabil, Ahmed Saeed, Esraa Mohamed
ID Number: 47899, 39119, 81774
Department: Accounting, IT, Manufacturing
Job Title: Vice President, Programmer, Engineer

The program should store this data in the three objects, then display the data for each employee on the screen. Finally write the data to a csv file.