Python Programming Assignment # 6

Instructions

For problems 1 and 2, create an error free efficient Python program including a main function calling other functions. Each program should be submitted in a separate Python file respectively that follows a particular naming convention. (E.g. The Python program for Problem 1 should be in .py file with name Assign6Answer1.py. The Python program for Problem 2 should be in .py file with name Assign6Answer2.py. The programs should execute properly in PyCharm). Problem 3 is on Huffman coding and answer needs to be submitted in a Microsoft Word document or PDF format.

Submit your assignment by Friday 7 December 2018 EoD.

Problems

Problem 1: Polar coordinates of a point (10 points)

Define a class named Point that will represent a point on a graph. To create a point pass the x and y co-ordinates to the constructor:

```
>>> p1 = Point(1,1)

>>> p2 = Point(4,5)
```

Include the following methods in your class (the examples refer to the two points p1 and p2 shown above):

• The __repr__ method should display the point in standard mathematical notation, e.g.

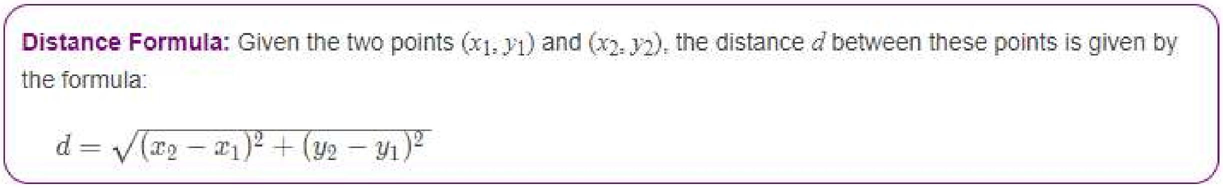
```
>>> p1

(1,1)
```

• A method named `dist` should compute the distance between two points, e.g.

```
>>> p1.dist(p2)

5.0
```



Distance Formula: Given the two points (x_1, y_1) and (x_2, y_2), the distance d between these points is given by the formula:

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

Read more: https://www.purplemath.com/modules/distform.htm

• A method named `polar` should return a pair of values corresponding to the polar coordinates of the point:

```
>>> p1.polar()
```

```
(1.4142135623730951, 0.7853981633974483)
```

The polar coordinates of a point (x, y) are a pair of numbers (r, q) where r = sqrt(x^2 + y^2) and and q = $\tan^{-1}$ y/x (Python's math library has a function named atan that computes $\tan^{-1}$). The formula for polar coordinates is valid only if the x-coordinate of a point is greater than 0. The polar method should return None if x is negative or 0.

Read more: http://tutorial.math.lamar.edu/Classes/CalcII/PolarCoordinates.aspx

Problem 2: String to decimal conversion (10 points)

Define a class named Conversion that will convert numbers from a particular format to decimal number. To initialize a converter object use the constructor:

```
>>> converter = Conversion()
```

Include the following methods in your class (the examples refer to the object converter shown above):

- The `roman_to_int` method should accept a string representing Roman number and convert it to decimal. e.g.

```
>>> print(converter.roman_to_int('MMMCMLXXXVI'))
3986
```

Use the following dictionary for conversion:

```
rom_val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
```

- The `binary_to_int` method should accept a string representing binary number and convert it to decimal. e.g.

```
>>> print(converter.binary_to_int('11100101'))
229
```

- The `octal_to_int` method should accept a string representing octal number and convert it to decimal. e.g.

```
>>> print(converter.octal_to_int('54'))
44
```

- The `hexadecimal_to_int` method should accept a string representing hexadecimal number and convert it to decimal. e.g.

```
>>> print(converter.hexadecimal_to_int('C1'))
193
```

Each of the above functions should check whether input contains a character that is outside the range of corresponding number system and should throw an exception in such cases. E.g.

```
>>> print(converter.octal_to_int('99'))
```

```
Invalid character: Exception: Invalid octal character in input string.
```

While calling the function, you should catch this exception as follows:

```
try:
   print(converter.octal_to_int(‘99’))
except Exception as e:
   print('Invalid character: ' + str(e))
```

Problem 3: Huffman Coding (10 points)

How many bits may be required for encoding the following message using Huffman coding? (Answer using the steps below)

Message -> ***Being deeply loved by someone gives you strength***

a. Complete the following frequency table for all characters in the message (White space ` ` is also a character).

b. Completely build the Huffman tree using the frequency table above.

c. Generate Huffman codes from the tree built in above step.

d. Write down encoded message and calculate total number of bits required to represent above message.

e. Considering that without encoding, it takes 1 byte to represent each character, what is the compression ratio achieved by Huffman coding for above message? Compression ratio is the ratio of number of bits required to represent encoded message and number of bits required to represent message without encoding.