



Instructions:

1. Upload answer of each question on its dropbox on Moodle.
2. No aids (such as calculators or search engine) are permitted.
3. All the questions are clear and no need to ask for more help.
4. All the question have the same marks.
5. You don't need to have any text file. However, in order to test your code, you can save the examples of each question in to a text file and test your codes based on that.

You have 3 hours to complete the exam. After exam, **NO** answer will be accepted through email.

1) Complete the **Python function** that begins with the header given below. The parameter filename is the name of the file to be read. The first few lines of the file might be:

06:45:31 01:12:59 00:00:00

10:10:10

15:19:31 13:21:17

Each line consists of one or more times of the form hh:mm:ss, where hh is the hour, mm is the minute and ss is the second. The hour, minute and second are separated from each other by a : . The times are separated from each other by **whitespace**. There may be more than one space separating the times on a line. Read the lines from the file and create lists of integer hours, integer minutes and integer seconds from the data in the file. **Return the lists of hours, minutes and seconds in that order.**

```
def readFile(fileName):
```

2) Complete the Python function that begins with the header given below. This function will be used in question 3. This function is given a **list** of words and **returns a dictionary** containing the unique words and the number of times each unique word occurs in the list of words, which is the frequency of the word. You may assume all words contain only lowercase letters.

For example if the list of words is ['this', 'is', 'a', 'test', 'this', 'is', 'not', 'mine', 'is', 'it'], then the dictionary should contain {'this':2, 'is':3, 'a':1, 'test':1, 'not':1, 'mine':1, 'it': 1}.

Return the **dictionary** of unique words and their frequencies.

```
def unique(words):
```

3) Write a **complete Python program** that calls the functions `readfile` from question 1 and `unique` from question 2, you **have to rewrite** (copy and paste) those functions here in order to show a correct output for this question. The program must do the following:

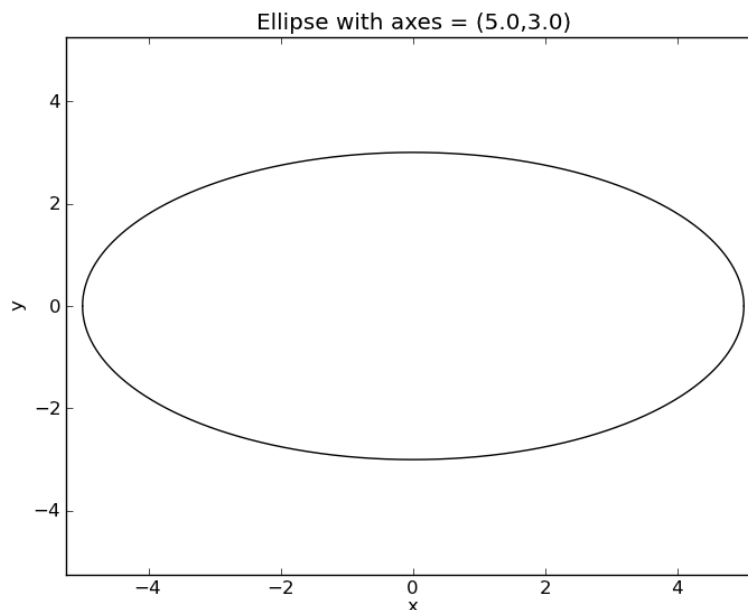
- Input the name of a file as a string.
- Loop as long as the name of the file is not an empty string.
- Call `readfile` to get the list of words from the file.
- Call `unique` to get the dictionary of unique words and their frequencies.
- Print a heading as shown in the sample run of the program.
- Print each word and its frequency from the dictionary on a separate line as shown in the sample run of the program. The order in which this information is printed is based on the order of the words in the dictionary.
- Input the name of the file as a string.

A sample run of the program follows:

Enter the name of the text file: data.txt

Word	Frequency
a	1
this	2
my	2
car	1
is	3
test	1
house	1
sentence	1
that	1

4) Complete the **Python function** that begins with the header given below. This function computes the X and Y coordinates of points on the top half of an ellipse, where the X coordinates range from **-a to a**, twice the length of the semi-major axis, and the Y coordinates range from **0.0 to b**, the length of the semi-minor axis. In the diagram shown below **a = 5.0** and **b = 3.0**.



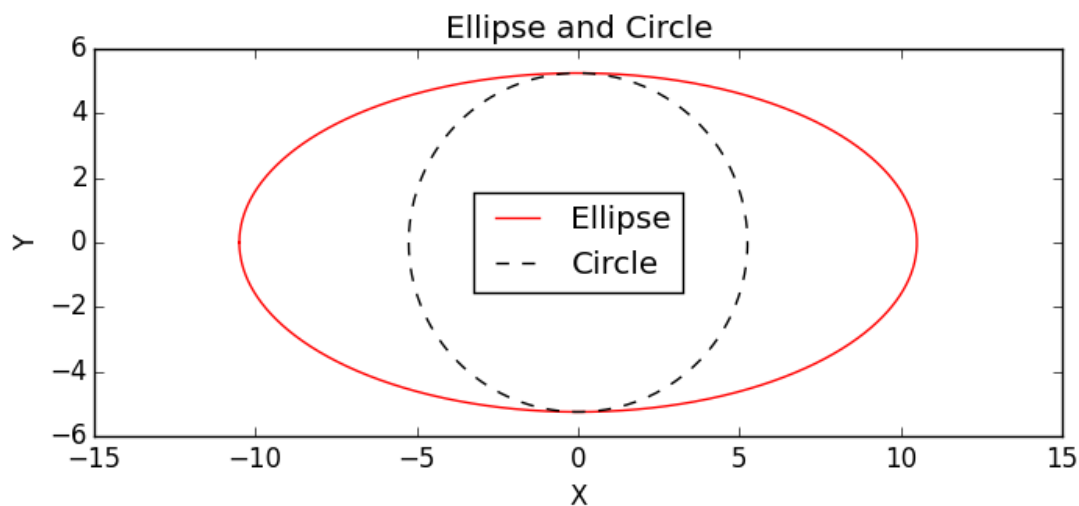
Create an array of equally spaced X coordinates from **-a to a** based on the number of intervals given to the function. Do this using one of the functions that create arrays. Using equation 1 and vector arithmetic create an array of Y coordinates that contains one Y coordinate for each X coordinate. There must not be any loops in this function!

$$y = b \times \sqrt{1 - \left(\frac{x}{a}\right)^2} \quad (\text{equation 1})$$

Return the array of X coordinates followed by the array of Y coordinates.

```
def ellipse(a, b, intervals):
```

5) Complete the **Python function** that begins with the header given below to display two curves in a single figure. Use functions from *matplotlib* to create the plots. Specify red as the colour of the first curve to plot and black as the colour of the second curve to plot. Use the arrays *xCoords1* and *yCoords1* for the first curve and the arrays *xCoords2* and *yCoords2* for the second curve. Use *legend1* for the first curve and *legend2* for the second curve, and position the legend as shown below. Label the axes as shown below. Use *legend1* and *legend2* to create the title and the name of the file to save. Save the figure as a .png file. Be sure to display the plot.



NOTE: You can use the formula of ellipse from question 4.

NOTE: Formula for a circle is: $x^2+y^2=r^2$; you may put appropriate *r* value from the figure.

```
def plotCurves(xCoords1,yCoords1,legend1, xCoords2,yCoords2,legend2):
```