Problem 1: How to calculate a home's value

Housing economists have long used a home price/rent ratio as one way to gauge whether or not home prices are inflated or undervalued.

A housing P/E

The use of a price/rent ratio is analogous to employing a price/earnings ratio for stocks. When a stock price is high, and its earnings per share relatively low, the P/E is high. A high P/E often indicates that the stock is too expensive, and the share price is headed for a drop.

What someone is willing to pay to rent a place is that home's "earnings." And, just as in the stock market, a high home price related to the rental earnings mean homes values will probably drop.

For a specific look at how a home's P/E is determined, let's consider a home that is listed for either rent or sale in suburban Chicago.

The home has been rented for the past three years for $1,600 per month; so that is $19,200 per year. It is currently listed for sale at $400,000. Dividing the price by the total annual rent of $19,200 gives a "housing P/E" of 20.83. According to Moody's Economy.com, the long-run average housing P/E is 16, so a P/E of 20.83 suggests that this home may be somewhat overpriced.

For this programming project you will read in a list of housing prices and how much each house is rented for that are found in the attached HousingPrices.txt file. The file looks like this:

```
713047     1246

1605787    1979

1174719     1879

1018239    1700
```

Where the first value on each line is the price of the home, and the second number on each line is the price that this home currently rents for per month.

Read these numbers in from the file into 2 separate arrays of size 100, since there are 100 home prices/rents in the file. Do this part in your main method.

Then use a for-loop to loop through the arrays and for each pair of values send them into a method that you will write called "getHousingPE". Your method will have two parameters; the price of the home and the current monthly rental rate of the home. Your method will calculate and then return the homes “housing P/E”.

Check each homes returned P/E value and if the P/E value is greater than 16 you should write to the output “House # is somewhat overpriced”; where # is the number of the house in the file (1-100). If the P/E is greater than 19 then you should write to the output “House # is way overpriced”. If the P/E is less than 12 then you should write to the output "“House # might be a good deal”. And if the P/E is less than 9 then you should write “Buy House #, it is a deal”. If the P/E is between 12-16 then output "House # is average".

Problem 2: Determining Left or Right

For this programming project you are to read in a list of street addresses and determine how many of those addresses are on the right side of the street, and how many are on the left. The street addresses file is a text file that looks like:

```
6 S 33rd St
6 Greenleaf Ave
618 W Yakima Ave
74 S Westgate St
3273 State St
```

You can determine if an address is either on the left or right side of the street by looking at the number of the address; all even numbers are on the right side of the street, and all odd numbers are on the left side of the street.

You will read your data into 2 arrays; an integer array to hold the street number, and a String array to hold the rest of the address. These will be parallel arrays, meaning the street number and street address of each index are related to each other.

Remember to draw a picture so that you can visualize this.

You will need to write a method that will determine if each house is on the left or right side of the street.

You will call this method for every street address, and then keep a count of how many houses are on the left and right sides of the street.

The output should be the street number and address and then whether the house is on the left or right side of the street:

```
Address                        Left/Right
6649 N Blue Gum St              left
4 B Blue Ridge Blvd             right
8 W Cerritos Ave #54            right
639 Main St                     left
…
```

The number of houses on the left and right sides of the street should also be output.

Problem 2: Horse Weights

Research shows that horses are satisfied when they consume an amount equivalent to a certain percentage of their body weight on a daily basis. Depending on the horse type, some are satisfied with as little as 1.8% of their body weight, but others may require up to 3.2% of their body weight.

An average of 2.0% is a good starting point for most “maintenance” and “light work” equines. The intake level for “moderate work” equines is slightly higher at 2.5%. For “growing” equines, “lactating” equines or equines that are subject to “heavy work” start with a 3.0% intake level.

Given a list of horse weights and horse types, you are to make a program that should output the amount of feed that each horse needs.

Attached is a list of 100 horse weights and horse types in the following format, with each horse weight and horse type on a single line:

```
1300    moderate work
700     growing
1232    moderate work
1300    heavy work
1320    lactating
760     growing
…
```

Read in the data into 2 arrays of appropriate type. Then make a function that will have 2 parameters; a single horse’s body weight, and the horse’s type. The function should return the horse’s amount of feed.

You will then output the index number for each horse, 1-100, and how much feed that equine should receive. (Note that we will use pounds for the body weight, and therefore our amount of feed will also be in pounds.)

Here is an example of some output – using the sample input from above:

```
Horse #     Amount of Feed
   1           32.5
   2           21
   3           30.8
   4           39
```

```
5           39.6
6           22.8
```

Here is the rubric that will be used to grade these problems:

Criteria	Point Value
Formatted correctly; indented correctly, no extra comments or blank lines, name is at the top of the program with a date, class name is capitalized	10
Correct data types; choose the correct type of variable for your data, with properly named variables, using arrays when necessary	5
Function or method made correctly; function or method has correct parameters and return type and works properly	10
Correct use of arrays; for loops are used with the array and the array is used correctly in the program	5
Algorithm is created at the top of the program and proper verbiage and indentation is used	10
Read data from the text file as input; the data is properly opened and checked for errors, and the data is read into the program and used	10
The program works as specified; the correct input is used and correct output is produced, and the mathematical function is implemented correctly	50
TOTAL	**100**
Extra Credit: Data is written to a text file, as opposed to being just written to the console screen	15
TOTAL with Extra Credit	**115**

You are to use the text files that are given, do not modify these text files.

Turn in your completed program to the Assignment in Blackboard; you will only need to turn in your program file.