

## **Junior Knights: AP Review - Practice Programs (If, Loops, ArrayLists, Static Methods)**

1) Write a program that calculates the amount of money stored in a bank account after a certain number of years, using simple interest calculated annually. The user should enter in the initial amount deposited into the account, along with the annual interest percentage and the number of years the account will mature. The output should provide the amount of money in the account after every year. (If you don't know how simple interest works, please look it up online or consult an Algebra I textbook.)

2) Write a program to take in a positive integer  $n > 1$  from the user and print out whether or not the number the number is a perfect number, an abundant number, or a deficient number. A perfect number is one where the sum of its proper divisors (the numbers that divide into it evenly not including itself) equals the number. An abundant number is one where this sum exceeds the number itself and a deficient number is one where this sum is less than the number itself. For example, 28 is perfect since  $1 + 2 + 4 + 7 + 14 = 28$ , 12 is abundant because  $1 + 2 + 3 + 4 + 6 = 16$  and 16 is deficient because  $1 + 2 + 4 + 8 = 15$ .

3) Write a static method, `numInversions`, that takes in an `ArrayList` of `Integer`, which stores distinct values, and calculate the number of inversions in that array. An inversion in a list is when a bigger value comes before a smaller value. The list 3, 6, 2, 1, 5, 4 has the following 8 inversions: (3, 2), (3, 1), (6, 2), (6, 1), (6, 5), (6, 4), (2, 1), and (5, 4). Then, write a main method to test your method `numInversions`. The method signature is provided below:

```
public static int numInversions(ArrayList<Integer> list) {  
    // Fill in method.  
}
```

4) Write a static method, `average`, that takes in an `ArrayList` of `Integer`, which stores test scores and calculates the average of those scores. Since the average may not be an integer, your method should return a `double`. Then, write a main method to test your method `average`. The method signature is provided below:

```
public static double average(ArrayList<Integer> list) {  
    // Fill in method.  
}
```

5) Write a static method, `max`, that takes in an `ArrayList` of `Integer`, which stores test scores and calculates the maximum of those scores. Then, edit the main method to test your method `average`. The method signature is provided below:

```
public static int max(ArrayList<Integer> list) {  
    // Fill in method.  
}
```

6) Write a static method, `min`, that takes in an `ArrayList` of `Integer`, which stores test scores and calculates the minimum of those scores. Then, edit the main method to test your method average. The method signature is provided below:

```
public static int min(ArrayList<Integer> list) {  
    // Fill in method.  
}
```

7) The standard deviation of a set of values,  $x_0, x_1, x_2, \dots, x_{n-1}$ , can be calculated as follows:

- 1) Find the average of all the values, let this value be  $A$ .
- 2) Calculate the sum of each term of the form  $(A - x_i)^2$ , for  $0 \leq i < n$ .
- 3) Divide the value calculated in step 2 by  $n$ .
- 4) Take the square root of the value calculated in step 3. This is your final answer!

Here is a quick run through with this sample list: 3, 6, 2, 5:

- 1) The average of the list is  $(3+6+2+5)/4 = 4.0$
- 2) The desired sum is  $(3 - 4)^2 + (6 - 4)^2 + (2 - 4)^2 + (5 - 4)^2 = 1 + 4 + 4 + 1 = 10$
- 3) Dividing by 4 yields 2.5.
- 4) The square root of 2.5 is roughly 1.58, the standard deviation of the list.

Write a static method, `stdev`, that takes in an `ArrayList` of `Integer`, which stores test scores and calculates the standard deviation of those scores. Then, edit the main method to test your method `stdev`. The method signature is provided below:

```
public static double stdev(ArrayList<Integer> list) {  
    // Fill in method.  
}
```

8) Integrate pieces 4 through 7 to write one coherent program that asks the user to enter in some test grades (terminated by a -1), and the prints out a summary of the exam results: Average, minimum, maximum, range, and standard deviation.

9) Read in a dictionary of words from the file "dictionary.txt" into an `ArrayList` of `String`. (First line of the file will store # of words, the rest of the words will follow, one per line, all lowercase, in alphabetic order.) Ask the user if they want to check a word. If they do, read in their word. Then, search their dictionary for the word using (a) linear search and (b) binary search. The search methods should have the following two signatures:

```
public static boolean searchLinear(ArrayList<String> words, String  
searchWord);
```

```
public static boolean searchBinary(ArrayList<String> words, String  
searchWord);
```