CS 170: Introduction to Computer Science I – Spring 2023
Homework Assignment #2

Due by Tuesday, February 21, 2023 at 4:00 pm

Submission instructions

Submit your assignment through the QTest system, using exam ID: **CS170.hw2**.
No email submissions are accepted. No late submissions are accepted.

General instructions and hints

In those problems asking you to write a method, always call the method several times to test that it works properly with a variety of different values for the parameters. Test it on more examples than the ones shown in this handout. Even if the problem asks you for just one method, you can always write additional helper methods to simplify and organize your code. Make sure you write comments to explain your code.

Comment requirements: Always comment the top of each method (what the method does, the meaning of the input parameters, the meaning of the output value). Write comments within the methods to explain the strategy you are using to solve the problem, and to clarify blocks of code that may be difficult to understand.

Problem 1: Sum of squares (4 points)

Write a method `sumOfSquares` which takes a positive integer `n` and returns the sum of the squares from 1^2 up to n^2. For example, `sumOfSquares(4)` returns 30, since 1^2 + 2^2 + 3^2 + 4^2 = 1 + 4 + 9 + 16 = 30. Do not use a formula to do this. Instead, add each term in a “for” loop.

Rubric for problems 1-10:
Programs that do not compile get zero points.
+4 correct implementation (up to 2 points for a partially correct solution. Zero points if the solution is far from correct.)
-1 incorrect method signature (method name, number of parameters, types of parameters, or return type)
-1 if there are fewer than 2 test cases
-1 if there are no comments, insufficient comments, or bad usage of comments

Problem 2: Product (4 points)

Pretend that you don't have the multiplication operator available. Write a method `product` that takes two integers `x` and `y` and returns their product.

Problem 3: Compound interest (4 points)

Write a method `compoundInterest(double money, double interestRate, int years)` that takes an amount of money, a yearly interest rate, and a number of years, and returns the total amount of money after that number of years. For example, `compoundInterest(1000, 0.05, 3)` returns 1157.625 (it's the total amount of money that you get if you deposit 1000 dollars with an interest rate of 5% for 3 years). Do not use `Math.pow()` to do this. Instead, add each term in a “for” loop.

Problem 4: PolySpiral length (4 points)

Write a method `polySpiralLength(int n, double base, int rounds)` that returns the total length of a polyspiral like those you implemented in a previous homework.

Problem 5: Decimal to binary (4 points)

We commonly represent integer numbers as sequences of 10 possible digits: 0, 1, 2, ..., 9. It is also possible to represent integers as sequences of only two digits: 0 and 1. Indeed, this binary representation is the way numbers are physically stored inside a computer's memory. Read more about binary numbers on Wikipedia: http://en.wikipedia.org/wiki/Binary_number
This web page also describes a procedure to convert a decimal number into a binary number.

Write a method `decimalToBinary` that takes an integer `n` between 0 and 255 (you can assume the method will always be called with a number between 0 and 255). The method converts that integer into a string containing an 8-bit binary representation of that number, i.e., a string with exactly 8 symbols (either 0 or 1). For example, `decimalToBinary(23)` returns the string `"00010111"`.

Problem 6: Ordered word (4 points)

Write a method named `isOrdered` that takes a string as input parameter, and returns `true` if that string's characters are sorted naturally in English alphabetical order (ignoring capitalization), or `false` otherwise. For example, your method should return `true` for input strings like "effort" and "Aaaabbbyy", whereas "Hello" should return `false` since the letter H appears in the English alphabet after the letter E.

Problem 7: Count vowels (4 points)

Write a method `countVowels` that takes a string `s` and returns the total number of vowels (A, E, I , O, U, both uppercase and lowercase) in the string.

Problem 8: Letter count (4 points)

Write a method `letterCount(String s, String c)` which takes a string `s` and a letter `c` (which is also a string of length 1), and returns the number of times the given letter occurs in the given string. You may assume the length of the letter is exactly 1, but the first string may be empty (of zero length).

Examples:
`letterCount("abc-123-abc-ABC", "b")` returns 2
`letterCount("This is interesting!", "z")` returns 0
`letterCount("", "q")` returns 0

Problem 9: Digit count (4 points)

Write a method `digitCount(int number, int digit)` which takes an integer (it could be negative) and a digit (between 0 and 9), and returns the number of times the given digit occurs in the given integer. You cannot convert the numbers to strings here (so, for example, you cannot use the `letterCount` method you just wrote). Solve this problem numerically. You can ignore leading zeros (for example, 001234 is the same as 1234).

Examples:
`digitCount(456234, 8)` returns 0
`digitCount(456234, 5)` returns 1
`digitCount(456234, 4)` returns 2
`digitCount(-456234, 4)` returns 2
`digitCount(0, 0)` returns 1

Problem 10: Word count (4 points)

Write a method `wordCount` which takes a string and returns the number of words in it. You may assume that the string contains only letters, digits, punctuation, or spaces. You may also assume that words are always separated by spaces. If two words are separated by multiple spaces, they still are only two words (so you cannot just count the number of spaces!). Everything that is not a space, including digits and punctuation, is considered part of a word. You are not allowed to use the method `split()` which is available for `String` objects in Java.

Examples:
`wordCount("This is a test")` returns 4
`wordCount("This   is   a    test!!!!!")` returns 4
`wordCount("")` returns 0
`wordCount("Wahoo42!??!?!")` returns 1

Problem 11: Let's buy a new car (5 points)

You want to buy a new car, and you are thinking about two alternatives, car A and car B. You like both cars, and you are considering keeping whichever one you buy for many years to come. As a result, you want to evaluate the total cost of ownership of both cars over many years. Car A costs $20,000, whereas car B costs $30,000. Car A runs 25 miles with a gallon of gas, whereas car B runs 32 miles with a gallon of gas. You estimate that you will drive about 15,000 miles each year, and that gas price is going to be around $2.50 per gallon for the foreseeable future. The maintenance cost of car A is $1,300 in the first year, then that cost increases 15% (with compound growth) each year. The maintenance cost of car B is $1,000 in the first year, then that cost increases 10% (with compound growth) each year.

Task 1. Write a method named `compareCars(int years)` that takes an integer number of years and **prints** a table with the total ownership cost of each car for each year, rounded to the nearest dollar, starting from year 0 which represents the time of purchase (see example for the table format). The method will **return** the name of the more economical car on the last year. Note: For this problem, you are allowed to break the usual “no print statements inside a method” recommendation.

For example, compareCars(2) will **print** the following table:

```
Year   Car A  Car B
0      20000  30000
1      22800  32172
2      25795  34444
```

and will **return** the String `"Car A"`.

Task 2. Using your method, answer this question: which car would you buy if you plan on keeping it for 5 years? 10 years? Discuss the results. Write your answer in a comment at the top of your code.

Task 3. Answer this question: in a method, what is the difference between printing something and returning it? Write your answer in a comment at the top of your code.

Rubric:
Programs that do not compile get zero points.
+3 correct implementation (up to 2 points for a partially correct solution. Zero points if the solution is far from correct.)
-1 incorrect method signature (method name, number of parameters, types of parameters, or return type)
-1 if there are fewer than 2 test cases
-1 if there are no comments, insufficient comments, or bad usage of comments
+1 for a correct Task 2 answer (based on the results of their code) and justified answer (+0 points for incorrect answer or no justification)
+1 for an answer to Task 3 that displays an understanding of the difference between printing and returning (+0 points otherwise).

Problem 12: The storyteller (5 points)

Write a Java method that generates and returns a simple fairy tale such as the following:

Alice was a beautiful princess. Betsy was a wicked witch. Alice had 10 expensive dresses, whereas Betsy had only 5. Because of envy, Betsy cast a spell on Alice, yelling these arcane magical words: "Eaaa yeeb siid! EAAA YEEB SIID!". Suddenly, Alice's 10 dresses turned into rugs. Alice became very unhappy. Seeing that Alice was unhappy, Betsy realized she was wrong, and reversed the spell. Alice and Betsy became friends, and they lived happily ever after.

The story your program generates will have a similar structure, but should be personalized based on the arguments passed to the method. The name of your method should be `storyteller` and should take 7 parameters:

1) Name of the first girl (e.g., Alice)
2) Name of the second girl (e.g., Betsy)
3) Possession that causes envy (e.g., dresses)
4) Quantity of the object that causes envy (e.g., 10)
5-7) Other words that can be plugged in the story as you wish. Collect at least 3 additional pieces of information that you will correctly place in the story.

You can change the sentences of the story as you wish, but you must include these elements:

1) A magical sentence, constructed as follows: The first word is composed of the last letter of the first girl's name, followed by two “a” and the first letter of her name. The second word is composed of the last letter of the second girl's name, followed by two “e” and the first letter of her name. The third word is composed of the last letter of the possession that causes envy, followed by two “i” and the first letter of that object. The magical words are repeated twice, with the correct punctuation and capitalization as reported in the example.

2) The second girl should have half of the desirable objects that the first girl has.

Rubric:
Programs that do not compile get zero points.
+5 correct implementation:

- **1 point for correct use of names**
- **1 point for correct use of object and quantity of envy**
- **1 points for incorporating three additional pieces of information**
- **1 point for correctly constructed magical sentence**
- **1 point for second girl having half the desirable objects that the first girl has**

-1 incorrect method signature (method name, number of parameters, types of parameters, or return type)
-1 if there are fewer than 2 test cases
-1 if there are no comments, insufficient comments, or bad usage of comments

Bonus points: Early submission

If you submit the entire homework no later than 48 hours before the deadline, and the total score on the rest of this homework assignment is at least 20 points, you will receive 2 bonus points. The bonus points will be added to the total score of this homework assignment.

Good luck and have fun!