Loyola University Chicago

department of computer science

exam two

comp 170

**Name: Jeffrey Ricketts-Hagan**

**Date:  07/03/2021**

By taking this exam I agree to follow these terms:

0) I may not copy or distribute this exam to anyone.

1) I may not discuss this exam with other students. I may not seek assistance from anyone I can only access Sakai, and the normal course tools and text; **nothing else**.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Question | Grade | Possible |
| 1 | Program 1 – Perfect Numbers |  | 50 |
| 2 | Program 2 - Friends |  | 50 |
|  | TOTAL |  | 100 |

Note the above point values for each question; **use your time appropriately**, Time will not be extended. Be sure to allow time to save you work and turn in properly.

**HOW TO TURN IN YOUR WORK FOR THIS EXAM:** To turn in your programs, add two .java file to the Sakai assignment. Be sure you save your final version. DO NOT SUBMIT SCREEN SHOTS.

**Code Grading Reminders:**

* Code that does not compile without errors in the standard course tools will receive zero.
* Missing or incorrect key features or requirements will lose 20%.
* Minor errors, including formatting, will lose up to 10%
* Incorrect programming style will lose from 5 to 10%

**question 1: perfect numbers program**

{50 points} Write a well designed and structured program to find perfect numbers. A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. For example 28 is a perfect number since 1+2+4+7+14 = 28.

Your program must include a method with this signature:

public static void findPerfectNumbers( int howMany );

When called it prints out the first **howMany** perfect numbers from lowest to highest. Requirements:

* Think before programming – in addition to the findPerfectNumbers method you must have at least one more method used in the solution (use as many as you wish). Your design should be clear from comments (pseudo code) and/or carefully chosen variable and method names. Make your code readable!
* Call and test your program from main in some suitable way (sample calls to findPerfectNumbers or input from the user).
* You must search with code for the perfect numbers. Our prior work on factors should be helpful. DO NOT just look up and use a list of known perfect numbers to pick from!
* Testing notes – there are very few perfect numbers. 8128 is the 4th perfect numbers and the highest one less than 10,000. Your program should be able to run and find at least the first four; try for 5 or more if you are very patient. 33,550,336 is the highest perfect number less than 50 million. There are no known odd perfect numbers, but no one has been able to prove that none exist – if you find an odd one either your code has a bug or you can immediately apply for a dual PhD in math and computer science.

**Question 2: My Friends**

{50 points} Write a well designed and structured program to create a list of friends and then search it for names or parts of names.

Requirements:

* Prompt the user to enter a number between 5 and 50 (inclusive). If any other number is entered, note the error and continue (repeatedly) to prompt until the user enters a number in the correct range.
* Prompt the user to enter names of that number of their friends and store them (an array should be helpful to so this). Names should be treated as case sensitive (allow both upper and lower case)
* After all the names are entered, allow the user to search the list to find friends; allow any number of searches, one after the other, until the user enters DONE.
* For each search find all friends that match an arbitrary input search string. Find all the friends that partially match the search string and print them out. If no matches are found, say so.
* DON’T write all the code in main. Use at least two methods that pass information between them and main or each other for the implementation. Your design should be clear from comments (pseudo code) and/or carefully chosen variable and method names. Make your code readable!

Search example: if 3 names John Smith, Jane Smith, Anoh Garg have been entered, a search for “Smith” would display both John Smith and Jane Smith; a search for “oh” would display John Smith and Anoh Garg.