Program must be named: yourLastNameFirstLetterOfYourFirstNamepgm1.java
If your name is George Washington the program should be named: WashingtonGpgm1.java

Turn in the signed source code on paper, and email me the source code.

Make sure the program is properly documented and aligned uniformly, looking professionally, I will take points off if it is not.

Include the following header in every program:

```
/***************************************************************************/
Author     : Your Name
Course     : COP 3530 Date, Time and place of class
Professor  : Michael Robinson
Program #  : Program Purpose/Description
             {A brief description of the program }
Due Date   : MM/DD/YYYY
Certification:
I hereby certify that this work is my own and none of it is the work of any other person.
........{ your signature }........
/***************************************************************************/
```

Purpose of this program: Implement chapter 2 ideas

1 - Worth 3 points
From the main method call a method named BigO, in the BigO method do the following:
*** Using one set of two for loops (a nested for loop)
- Create a two dimensions array of ints of size 10 x 10
- Load all indexes with the addition of its row + column
- Display the two dimension array with its data displaying a perfect square
- Analyze the contents of the array and obtain the following totals:
  - Total for all values found in columns 0 and 9
  - Total for all values found in rows 4 and 5
  - Total for all values found in columns 5 and 4
  - Total for all values found in rows 9 and 0
- Print the big O of this method

2 - Worth 3 points
- Using ints only, implement recursion to find the factorial of 50
- Find out what is the first positive int where recursion fails
- Using longs only, implement recursion to find the factorial of 50
- Find out what is the first positive long where recursion fails
- Find the factorial of 50 using BigInteger data type
- Find out what is the first positive BigInteger where recursion fails
- Print the big O of each of these methods

3 - Worth 2 points
Using binary search
- Load into primary memory in a single dimension array the numbers from 1 to 1,000,000
- Find out how many reads it takes to find the numbers:
  15; 279,025; 775; and 897,000 e.g.
  e.i. : To find number x it took y reads.
- Print the big O of this method