Java Exam: File I/O, Arrays, Sorts Name _____ Per ___

A. Array short questions

Consider the array below, called Arr

5.6	7.1	12.1	6.7
0.5	-1.2	3.5	4.1
0.02	3.14	45.6	-12.6

1. Show with curly brackets how you would declare and initialize the array.

2. If you were to make a second array called Arr2 to accept all of the values from Arr plus two additional rows, show the single line of code to declare Arr2 and specify its size (do not use curly brackets).

3. Show nested loops to take all values from Arr and place them in the first 3 rows of Arr2. In your same nested loops, place 1.1 in the final two rows of Arr2.

4. Fill in the blanks below. If there is an error, specify what kind.
   a) Arr [3][1] returns _____ c) Arr [1][2] returns _____ e) _____ returns 4.1
   b) Arr.length returns _____ d) Arr[2].length returns _____

B. Sorts Questions

Write Bubble (B), Selection (S), or Insertion(I) to the left of each clue. Each blank can receive 1-3 terms. Each term can be used once, many times, or not at all.

_____ Before EVERY swap, only two items are compared.
_____ Utilizes a temporary variable to hold a value from the array.
_____ Is used with a 1-D Array.
_____ Is similar to holding an already-sorted hand of cards, receiving a new card, and putting the new card in the proper location, after moving cards down to make room.
_____ Starts by looking at the whole array to find the smallest/largest value, before a single swap is done.
_____ Starts by just looking at two parts of the array to decide if a swap should be done, but much later looks at the whole array to decide if a swap should be done.
_____ For an unordered array, is the slowest quadratic sort (of these three).

C. Program:

Write a program called CrimeSummary.java that:

- Reads file called “Cybercrime.txt”
- Creates a 2-D Array called Summ with selected information from the file
- Finds the average of years in jail for all like crimes
- Writes the crime type and average years in jail to “Summary.txt”

Each line of Cybercrime.txt if formatted like this:

[Crime]: [Description]. “Years in jail]: [Number]

Here’s the Cybercrime.txt file:

Espionage: Matt Cross who spied on Americans for the KGB using computerized radar conveyance technology. Years in jail: 20

Espionage: Joe Martin used a Scanner to read in military records to find aliases of soldiers overseas. Years in jail: 5

Fraud: Stew Wilson stole credit card numbers from Amazon and sold them on the black market. Years in jail: 3

Fraud: Reggie Taylor hacked into Bank of America and filed for 200 small loans under other names, whose accounts forwarded the money to Reggie. Years in jail: 10

Espionage: Frida Smith hacked into the Pentagon database and altered sensitive information, resulting in serious financial and life loss. Years in jail: 12

Your program should make a 2-D Array called Summ that looks like this:

Crime	Description	Years in Jail
Espionage	Matt Cross who spied on Americans for the KGB using computerized radar conveyance technology.	20
Espionage	Joe Martin used a Scanner to read in military records to find aliases of soldiers overseas.	5
Fraud	Stew Wilson stole credit card numbers from Amazon and sold them on the black market.	3
Fraud	Reggie Taylor hacked into Bank of America and filed for 200 small loans under other names, whose accounts forwarded the money to Reggie.	10
Espionage	Frida Smith hacked into the Pentagon database and altered sensitive information, resulting in serious financial and life loss.	12

Your program should write to Summary.txt (a brand new file) as shown below:

The average jail sentences in years for each crime:

Espionage: 12.3

Identity Theft: 6.5

Complete the code shown.

```
public class CrimeSummary {
```

```
static _____ [] [] Summ = new _____ [ ____] [ ____];

public static void main (String [] args){

        readFileNMakeArray();

        writeFile();

}

public static void readFileNMakeArray(){









        line = reader.nextLine();

        Summ[a][0]= line.substring(0, line.indexOf(':');

        Summ[a][1] = line.substring(line.indexOf(':')+1, line.indexOf('.')+1);

        Summ[a][2] = line.substring(line.lastIndexOf(':')+1, line.lastIndexOf('.');



}

public static void writeFile(){









}
```