CISC 1115 Fall 2018 Final Exam

Question 1. (12 points – 4 points each)

Each of the parts a. through c. below is preceded by a comment indicating what the code should do. There is a least one problem with each section of code and it fails to do what was intended. Show how to modify or rewrite the code the code so that it does work as intended. If there are multiple problems, correct each one. Assume that all variables used have already been declared.

a.
```
// INTENT: given an array arr of int values
// set small equal to the smallest of the array values

      small = arr[0];
      for (int j = 0; j < arr.length-1; j++)
          if (small < arr[j])
              arr[j] = small;
```

b.
```
// INTENT: compute the average of all n values in the integer array arr;
// compute the average as a double and store it in the double variable avg

      int sum = 0;
      int n = arr.length;
      for (int k = arr.length; k <= 0; k--)
           sum = sum + arr[k];
      double avg = sum / n;
```

c.
```
//INTENT: the array b should have the values of array a in reverse order
      int [ ] a = {1,2,3,4,5};
      int [ ] b = new int[5];
      for (int i=0; i<=a.length; i++)
            b[i-1] = a[a.length-i];
```

Question 2. (15 points)

What is printed by each of these pieces of code?

a.
```
int x = 5;
do {
     System.out.println(x);
     x = x + 3;
} while( x < 10 );
System.out.println("now "+ x);
```

b.
```
for (int i = 3; i > 1; i--) {
     System.out.println(i + " and " + (i * 10 - 1));
}
System.out.println("finished");
```

c.
```
double z = 5.4321;
int j = 4;
while (j < z) {

      System.out.println(j +" is less than " + z);
      j++;
}
System.out.println ("it is " +j +" at the end");
```

Question 3. (13 points)

What is printed by the following sections of code?

a.(5 points)

```
String source = "ID is 12-AB-1X ";
int pos = source.indexOf('-');
int pos1 = source.indexOf(' ',pos+1);
String str1 = source.substring(pos+1,pos1);
System.out.println(str1);
```

b.(5 points)

```
source = "from Denver,CO.";
int pos2 = source.lastIndexOf(',');
int pos3 = source.lastIndexOf(' ',pos2);
String str2 = source.substring(pos3+1,pos2);
System.out.println(source.substring(pos3+1,pos2));
```

c.(3 points)

```
System.out.println(str1.compareTo(str2)>0);
```

Question 4. (6 pts)

Perform the following conversions :

a. 101010 (base 2) to base 10 Answer _____

b. CA (base 16) to base 10 Answer _____

c. 10101011 (base 2) to base 16 Answer _____

d. 50 (base 10) to base 2 Answer _____

e. F8 (base 16) to base 2 Answer _____

f. Perform the following addition of two binary (base 2) numbers. Your answer should be a number in binary. Answer _____

101 + 110 = ?

Question 5. (12 points)

What is printed by the following code found in main of a Java program with the methods below?

```
int[] arr = {1,2,3,4,5};

doOne(arr[0]);
System.out.println(arr[0]);









doTwo(arr,1);
System.out.println(arr[1]);









doThree(arr);
System.out.println(arr[2]);
```

The program contains the following methods:

```
static void doOne(int n){
    n=n*10;
}

static void doTwo(int[] arr, int j){
    arr[j] = arr[j+2] *10;
}

static void doThree(int[] arr){
    for (int i=0; i<arr.length; i++)
        arr[i] = arr[i]*5;
}
```

Question 6. (12 points – 4 points each)

a. An array, **numbers**, of integers is filled with 100 random numbers whose values range from 10 to 199 (inclusive). You DO NOT have to write code to generate the random numbers. Write the code to count how many elements of the array are even and between 30 and 40 (inclusive).

b. The quadratic formula can be used to solve for the two values of x in a quadratic equation of the form $ax^2+bx+c=0$. Write an expression in Java that will compute one of the values of x based on the formula below:

$$\frac{-b+\sqrt{b^2-4ac}}{2a}$$

c. Suppose I am doing a binary search on the following array for the number 1

```
int nums[ ] =  {5, 77, 78, 89, 100, 117, 125, 235, 390, 1000};
```

List the high, low and midpoint values at each step until 1 is not found. DO NOT write any code. Enter into the table below the low, mid and high values as the binary search progresses. You may use either the subscript (index) values or the actual number stored at the subscript. Not all rows may be needed.

Low	Mid	High

Question 7. (30 pts)

Write a <u>complete</u> Java program with comments in main and in each method.

Data: The input data for this program is given as two columns of numbers. All data will be entered from a file named **`input.txt`** and all output will go to the screen. Assume there will not be more than 100 rows of data in the file.

<u>Sample Data Set:</u>

```
7 23.56
16 88.12
10 75.1
```

Design a **Java class** with a **main** method that does the following:

1) Reads the data into two arrays of doubles, `arr1` and `arr2,` by invoking the method `readData`**.** with both arrays as parameters. The return value from `readData` should be stored in an integer variable `num`.
2) Modifies all of the values in the `arr2` array by invoking the method `changeArray,` passing both arrays and `num` as parameters.
3) Sort `arr1` and `arr2` separately by invoking the method `sortArray` twice – once for `arr1` and once for `arr2`
4) Prints the values in both arrays to the screen in two columns as shown below with a header and the numbers right aligned

```
  arr1     arr2
  7.00   164.92
 10.00   751.00
 16.00  1409.92
```

Method Details:

The `readData` method reads two doubles values from the input file. The first value of each line is stored in `arrOne`, and the second value is stored in `arrTwo`. For example, in the Sample Data above, 7 would go into `arrOne[0]` and 23.56 would go into `arrTwo[0]`. This is repeated until there are no more values in the file.
Parameters:
`arrOne` - an array of <u>doubles</u> for storing the 1st value read in per row
`arrTwo` – an array of <u>doubles</u> for storing the 2nd value read in per row
Returns:
`num` - an integer representing the number of rows that were read in

The `changeArray method` has parameters:
`alpha` – an array of doubles
`beta` – an array of doubles
`k` - an integer representing the number of elements in `alpha` and `beta` to be processed

The method multiplies each element of `beta` by the corresponding value of `alpha` and stores the result into `beta.` For example, `beta[0],7,` is multiplied by `alpha[0], 23.56` and the result, `164.92,` is stored in `beta[0]`
.

The `sortArray` method sorts the first k elements of the parameter array into ascending order.
Parameters:
`arrToOrder` - an array of doubles to be sorted
`k` - an integer representing the number of elements in the array `arrToOrder`

Extra Work Page: