CISC 1115 (MY11) Exam 2 Spring 2024 Name: _____

Question 1. (20 pts)

What is printed by the following sections of code?

a.
```
String s = “Hello, How are you?”;
String s1;

s = s.substring(0, 15) + “we?”;
System.out.println(s);

s1 = s.substring(0, 4);
s = s.substring(5);
s1 += s + “\n”;
System.out.print(s1);

System.out.println(s1.compareTo(s) > 0);

if (s.indexOf(“Hello”) != -1){
    System.out.println(“Hello”);
} else {
    System.out.println(“Goodbye”);
}
```

b.
```
String first = “Monticello”;
String last = “NY”;
String middle = “Liberty”;
int k, m;
char c;

k = middle.length();
m = “Liberty”.length();
System.out.println(k + “ “ + m);

middle += last;
last = “Fallsburg”;
first = first + “N”;
first = first + “Y”;

System.out.println(first + “ “ + middle + “ “ + last);

if (first.compareTo(middle) > 0) {
   System.out.println(“bigger”);
} else {
    System.out.println(“smaller”);
}

c = (char) (middle.charAt(5) + 1);
middle = middle.substring(0, 5) + Character.toString(c) + middle.substring(6);

System.out.println(middle);
```

Question 2. (10 pts)

a. The following method is supposed to **check whether any of the values in the array are negative**. However, it doesn't work properly. Explain the problem and show how to fix the problem so that the code works properly.

```
public static boolean hasNegValue (int [] a, int n) {
        for (int i = 0; i < n; i++) {
            if (a[i] < 0) {
                return true;
            } else {
                return false;
        }
}
```

b. What is wrong with the following program segment? Will an error message be produced? How can this be corrected?

```
int [] b = new int [10];
for (int i = 1; i < 10; i++) {
    a[i] = i;
    System.out.println(a[i]);
    a[i] = a[i] + 1;
    a[i + 1] = a[i];
}
for (int j = 0; j < 10; j++) {
    System.out.println(a[i]);
}
```

Question 3. (20 pts)

Given the array of numbers below:

int [] nums = {12, 77, 87, 89, 100, 117, 125, 189, 235, 529, 1000};

Assuming that we are looking for the key of **235,**

a. Show the sequence of numbers you would encounter in this search if you used a **sequential search**.

b. Show the sequence of numbers you would encounter in this search if you used a **binary search**.

Enter into the table to the right the low, mid and high values as the binary search progresses.

DO NOT write any code.

You may use either the subscript (index) values or the actual number stored at the subscript. Not all rows of the table below may be needed.

Low	Mid	High

Question 4. (20 pts)

Write the Java code for **a method named maxChar()** that **receives two char arrays of the same size as arguments** and **returns a newly created array of type char**.

Each element of the new array that is returned is the larger of the corresponding elements of the two arrays received.

For example, if {‘x’, ‘3’, ‘?’} and {‘7’, ‘4’, ‘e’} were passed to maxChar(), it would return the array: {‘x’, ’4’, ‘e’}.

Or, if {‘6’. ‘f’, ‘n’} and {‘S’, ‘u’, ‘n’} were passed to maxChar(), then {‘S’, ’u’, ’n’} would be returned.

Question 5. (30 pts)

Write a complete Java program to **create an array to hold 10 randomly generated integers between 0-9** and **count how many times each number appears in the array**.

For example, if the array contains: **1 1 2 3 1 2 0 8 9 8**, the program should print:

Number of 0s: 1
Number of 1s: 3
Number of 2s: 2
Number of 3s: 1
Number of 4s: 0
Number of 5s: 0
Number of 6s: 0
Number of 7s: 0
Number of 8s: 2
Number of 9s: 1

In addition to your main method, you will write 2 methods:

- **printArray()**: prints the values in an array of integers (numList). This method takes one parameter, an array of integers, and returns nothing.
- **countNumbers()**: counts how many times each integer (0-9) appears in an array. This method takes one parameter, an array of integers, and returns an array of each count, where each index in the new array should correspond to the value that we are counting. Print these results to the console (you can do this in main or make a separate method)

  So if the array contains: **1 1 2 3 1 2 0 8 9 8**, the array created in countNumbers() should resemble the following:

Value:	**1**	**3**	**2**	**1**	**0**	**0**	**0**	**0**	**2**	**1**
Index:	0	1	2	3	4	5	6	7	8	9

Answer Question 5 Below: