1 Short Answer (15 Points Each)

1. For the following one-dimensional array, show the final array state after each pass of the three sorting algorithms.

   Solution:

   Bubble Sort:

   Original Array:

15	3	2	8	7	10	5

   ```
   Pass 1: 3  2  8  7  10  5  15
   Pass 2: 2  3  7  8  5  10  15
   Pass 3: 2  3  7  5  8  10  15
   Pass 4: 2  3  5  7  8  10  15
   Pass 5: 2  3  5  7  8  10  15
   Pass 6: 2  3  5  7  8  10  15
   ```

   Insertion Sort:

   Original Array:

15	3	2	8	7	10	5

   ```
   Pass 1: 3  15  2  8  7  10  5
   Pass 2: 2  3  15  8  7  10  5
   Pass 3: 2  3  8  15  7  10  5
   Pass 4: 2  3  7  8  15  10  5
   Pass 5: 2  3  7  8  10  15  5
   Pass 6: 2  3  5  7  8  10  15
   ```

   Selection Sort:

   Original Array:

15	3	2	8	7	10	5

   ```
   Pass 1: 5  3  2  8  7  10  15
   Pass 2: 5  3  2  8  7  10  15
   Pass 3: 5  3  2  7  8  10  15
   Pass 4: 5  3  2  7  8  10  15
   Pass 5: 2  3  5  7  8  10  15
   Pass 6: 2  3  5  7  8  10  15
   ```

2. Write a method that takes in two integer parameters, `size` and `n`, and returns a one-dimensional array of integers of size `size` and populated with random integers between 1 and n inclusively.

```java
public static int[] populate(int size, int n) {
    int[] A = new int[size];
    for (int i = 0; i < size; i++)
        A[i] = (int) (Math.random() * n) + 1;

    return A;
}
```

3. Write a method that takes in a single parameter of a two-dimensional array of doubles and returns a copy of the array. The copy is to be stored in a different memory location than the original.

```java
public static double[][] CopyArray(double A[][]) {
    double[][] B = new double[A.length][A[0].length];

    for (int i = 0; i < A.length; i++)
        for (int j = 0; j < A[0].length; j++)
            B[i][j] = A[i][j];

    return B;
}
```

4. Write a method that takes in a single parameter of a two-dimensional array of doubles and returns a one-dimensional array of doubles holding the row sums of the two-dimensional array. That is, the first entry of the one-dimensional array is the sum of the first row of the two-dimensional array, the the second entry of the one-dimensional array is the sum of the second row of the two-dimensional array, and so on.

```java
public static double[] RowSums(double A[][]) {
    double[] B = new double[A.length];

    for (int i = 0; i < A.length; i++)
        for (int j = 0; j < A[0].length; j++)
            B[i] += A[i][j];

    return B;
}
```

2 Program Traces (15 Points Each)

1. Determine the outputs for each of the following inputs.

```
public class Thing {
    private int x;
    private int y;
    private int z;

    public Thing() {
        x = 1;
        y = 2;
        z = 3;
    }

    public Thing(int a, int b, int c) {
        x = a;
        y = b;
        z = c;
    }

    public void PrintXYZ() {
        System.out.println(x + " " + y + " " + z);
    }

    public int DoSomething(Thing thing1) {
        int a = x * thing1.x;
        int b = y * thing1.y;
        int c = z * thing1.z;
        return a + b - c;
    }

    public int DoSomething(Thing thing1, int t) {
        if (t > 0) {
            while (t > 0) {
                x += t;
                t -= 5;
                thing1.z += z;
                PrintXYZ();
                thing1.PrintXYZ();
            }
        } else {
            for (int i = 0; i < -t; i++) {
                thing1.y++;
                z += i;
                PrintXYZ();
                thing1.PrintXYZ();
            }
        }
        return DoSomething(thing1);
    }
}
```

```
import java.util.Scanner;

public class Exam3Trace1 {

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        System.out.print("Input: ");
        int f = kb.nextInt();
        int g = kb.nextInt();
        int h = kb.nextInt();
        int p = kb.nextInt();
        int q = kb.nextInt();
        int r = kb.nextInt();
        int s = kb.nextInt();

        Thing thing1 = new Thing(f, g, h);
        Thing thing2 = new Thing(p, q, r);

        thing1.PrintXYZ();
        thing2.PrintXYZ();
        System.out.println();

        System.out.println(thing2.DoSomething(
            thing1));
        System.out.println();

        System.out.println(thing1.DoSomething(
            thing2, s));
    }
}
```

(a) `Input: 1 3 5 2 6 4 16`

```
1 3 5
2 6 4

0

17 3 5
2 6 9
28 3 5
2 6 14
34 3 5
2 6 19
35 3 5
2 6 24
-32
```

(b) `Input: 1 1 3 2 5 3 -3`

```
1 1 3
2 5 3

-2

1 1 3
2 6 3
1 1 4
2 7 3
1 1 6
2 8 3
-8
```

2. Determine the outputs for each of the following inputs.

```
import java.util.Scanner;

public class Exam3Trace2 {

    public static void PrintArray(int[][] A, int rows, int cols, int width) {
        String widthstr = "%" + width + "d";
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++)
                System.out.printf(widthstr, A[i][j]);
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        System.out.print("Input: ");
        int a = kb.nextInt();
        int b = kb.nextInt();
        int r = kb.nextInt();
        int s = kb.nextInt();

        int[][] Arr = new int[a][b];

        for (int i = 0; i < a; i++)
            for (int j = 0; j < b; j++)
                Arr[i][j] = i * j + (i + j % 2);

        PrintArray(Arr, a, b, 5);
        System.out.println();

        for (int j = 0; j < b; j++)
            Arr[a - 1][j] = Arr[a - 1][j] + Arr[a - 2][j];

        PrintArray(Arr, a, b, 5);
        System.out.println();

        for (int i = 0; i < a; i++)
            Arr[i][r] = Arr[i][r] + Arr[i][s];

        PrintArray(Arr, a, b, 5);
    }
}
```

(a) `Input: 3 4 1 2`

```
    0    1    0    1
    1    3    3    5
    2    5    6    9

    0    1    0    1
    1    3    3    5
    3    8    9   14

    0    1    0    1
    1    6    3    5
    3   17    9   14
```

(b) `Input: 2 5 0 3`

```
    0    1    0    1    0
    1    3    3    5    5

    0    1    0    1    0
    1    4    3    6    5

    1    1    0    1    0
    7    4    3    6    5
```

3 Coding (20 Points)

Write a program that will take three input integers from the user, a, b, and n. The main program will then create a one-dimensional array of integers of size n and put the values of a and b into the first two entries, in that order. The main program must then call a method `fib` that will take in a one-dimensional array as its only parameter. This function will then populate the rest of the array by setting each entry to the sum of the two previous entries. The main will then call another method that will print out the contents of the array. So if the user inputs 1, 4, and 7 the final array would look like.

1	4	5	9	14	23	37

Now have the main program call a method `ratios` on the array of integers that was just created. This method should take in a single parameter one-dimensional array of integers and return a one-dimensional array of doubles. The array of doubles are the ratios of the consecutive entries in the array of integers. So in our above example, the ratios would be $\frac{4}{1}$, $\frac{5}{4}$, $\frac{9}{5}$, $\frac{14}{9}$, $\frac{23}{14}$, and $\frac{37}{23}$. Note that there is one less ratio than there are entries in the array of integers. Create a method to print out the array of doubles and have the main program call this method on the ratio array. The output of an example run is,

```
Input: 1 4 7
1 4 5 9 14 23 37
4.0 1.25 1.8 1.5555555555555556 1.6428571428571428 1.608695652173913
```

```java
import java.util.Scanner;

public class Exam3Prog {

    public static void Print(int[] A) {
        for (int i = 0; i < A.length; i++)
            System.out.print(A[i] + " ");

        System.out.println();
    }

    public static void Print(double[] A) {
        for (int i = 0; i < A.length; i++)
            System.out.print(A[i] + " ");

        System.out.println();
    }

    public static void fib(int[] A) {
        for (int i = 2; i < A.length; i++)
            A[i] = A[i - 1] + A[i - 2];
    }

    public static double[] ratios(int[] A) {
        double [] B = new double[A.length - 1];

        for (int i = 0; i < A.length - 1; i++)
            B[i] = (double)A[i + 1] / A[i];

        return B;
    }


    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        System.out.print("Input: ");
        int a = kb.nextInt();
        int b = kb.nextInt();
        int n = kb.nextInt();

        int[] A = new int[n];
        A[0] = a;
        A[1] = b;

        fib(A);
        Print(A);

        double [] R = ratios(A);
        Print(R);
    }
}
```