Exam: Section B

Instructions

This section will involve writing code on a computer, not necessarily your own. Instructions about the computer will be provided at the exam.

- Note: you may NOT access the module materials, nor any other source such as books, other material on the internet and so on.
- Make sure that each program you write has the correct name as specified in each question.
- You can ask for blank paper to sketch out solutions but must return the paper with your NAME on it.

Exam questions

NOTE: the actual exam will have fewer questions. We are providing additional questions to help you prepare by practicing.

1. In `NumberTriangle.java` use for-loops to print

```
1
22
333
4444
55555
```

2. In `EvenList.java`

```
    public static void main (String[] argv)
    {
        printEvenList (8);
    }
```

write a method called `printEvenList()` that, given a number like N (for example N=8) prints the following:

```
Even numbers between 0 and 8:  0 2 4 6 8
Even numbers between 2 and 8:  2 4 6 8
Even numbers between 4 and 8:  4 6 8
Even numbers between 6 and 8:  6 8
Even numbers between 8 and 8:  8
```

3. In `Largest5.java`, write a method called `largestOfThree` and use it twice in main to find the largest of 5 numbers:

```
public static void main (String[] argv)
{
    int a=3, b=1, c=0, d=5, e=2;
    // Call largestOfThree twice here:

    System.out.println (largest);
}

static int largestOfThree (int x, int y, int z)
{
    // ... write your code here ...
}
```

4. In `DivisibleExercise.java`, copy over only those elements of an array that are divisible by 5.

```
public static void main (String[] argv)
{
    int[] A = {1, 4, 5, 10, 11, 15, 12, 10};
    int[] B = copyOverDivisibleBy5 (A);

    System.out.println (Arrays.toString(B));
    // Should print 5, 10, 15, 10
}
```

5. In `Twice.java`, Find the numbers in A that occur at least twice in B.

```
// Print numbers in A that occur at least twice in B.
int[] A = {5, 7, 11, 13, 17};
int[] B = {4, 5, 7, 3, 2, 7, 6, 7, 11, 11, 5};
int[] C = processArrays (A, B);
System.out.println (Arrays.toString(C));
// Should print 5, 7, 11
```

6. In `ArraySort.java`, complete the following to sort the array so that it's sorted in order of last to first in letter order:

```
char[] letters = {'g','r','e','e','t','i','n','g','s'};

// Your code here.

System.out.println (Arrays.toString(letters));
// Should print: [t, s, r, n, i, g, g, e, e]
```

7. In `OddArray.java`, add code to make an array of consecutive odd numbers:

```
public static void main (String[] argv)
{
    int[] odds = makeConsecutiveOdds (5);
    System.out.println (Arrays.toString(odds));
    // Should print [1, 3, 5, 7, 9]
}
```

8. In `Suffix.java`, add code below to determine if one word is a suffix of another:

```
public static void main (String[] argv)
{
    System.out.println (isSuffix("ran","bran"));
    System.out.println (isSuffix("ran","brain"));
    // Should print true and false
}
```

9. In `Multiple.java` write a method called firstMultiple3 that takes a number N and finds the first multiple of 3 whose square exceeds N.

```
public static void main (String[] argv)
{
    System.out.println (firstMultiple3(100));   // Should print 12
    System.out.println (firstMultiple3(1000));  // Should print 33
}
```

10. In `Array2D.java`, write code in arraySum2D() to sum the numbers across a particular row and down a particular column

```
public static void main (String[] argv)
{
    int[][] A = {
        {1,2,3,4,5},
        {2,3,4,1,6},
        {3,4,1,2,7},
        {4,1,2,3,8}
    };
    System.out.println (arraySum2D(A,1,2));
    // Should print 26
}
```