Vanier College
Computer Science Department

Programming 2

LAB 2

Q1: What is the output of the following code?

a)

```
public static void main(String[] args) {

 String[][] animals = {
 { "dog", "cat", "fish", "bird", "worm" },
 { "lion", "baboon", "bison", "beaver" },
 { "bear", "bat", "ant", "bobcat", "buffalo", "elephant"},
 { "crab", "coyote", "cow", "frog", "goat","grissly" }};


 for (int i = 0; i < animals.length; i++)
 {
      System.out.print("["+i+"]["+0+"]" + animals[i][0] + " --");

      for (int j = 1; j < animals[i].length; j++)
      {
        System.out.print("["+i+"]["+j+"]" + animals[i][j] + " ");
      }
      System.out.println();
      }
 }
```

b)

```
char ch = 'A';
char[][] ary = new char[5][];

for (int k = 0; k < ary.length; k++)
{
      ary[k] = new char[k+1];
      for (int j = 0; j < ary[k].length; j++)
            ary[k][j] = ch++;
}

//see the content
for (int k = 0; k < ary.length; k++) {
      for (int j = 0; j < ary[k].length; j++)
      {
            System.out.print(ary[k][j]);
      }
      System.out.println();
}
```

c)

```
public class Array {

 Array() {
 String[] ary = new String[6];
 load(ary);
 dump(ary);
}

 void load(String[] ary) {
  for (int n=1; n<ary.length; n+=2)
  {
        ary[n-1] =" Pos: " + n;
        ary[n]  =" Neg: " + (-n-1);

  }
 }

 void dump(String[] ary) {
 for (int n=0; n<ary.length; n++)
  System.out.println(ary[n]);
 System.out.println();
 }

 public static void main(String[] args) {
  new Array();
 }
}
```

Q2: Write a program to add the elements on the two diagonals of a square two-dimensional integer array and display that sum. Ensure that if there is a middle element in the array it is not counted twice in the sum.

For example, with the array:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

your program should display 68 (1+6+11+16+4+7+10+13).

With the array:

1	2	3
5	6	7
9	10	11

your program should display 30 (1+6+11+3+9) (note that the 6 is not counted twice).

Q3: A magic squares is an N-by-N matrix of the integers, such that all row, column, and diagonal sums are equal. For example,

4	9	2
3	5	7
8	1	6

is a magic square, because 4+9+2=15 3+5+7=15 8+1+6= 15 and 4+38=15
9+5+1=15 2+7+6=15 and 4+5+6=15 2+5+8 = 15.

Write a Java program to test if a matrix represents a magic square.

Q4: One way to generate a magic square of size n, when n is odd is to assign the integers 1 to n^2 in ascending order, starting at the bottom, middle cell. Repeatedly assign the next integer to the cell adjacent diagonally to the right and down. If this cell has already been assigned another integer, instead use the cell adjacently above. If the new column is outside the square start back at the first column. If the new row is outside the square, start back at the beginning of the row.

Write a Java program to generate a magic square of a given odd size. For example, if the user enters 3, you should generate:

4	9	2
3	5	7
8	1	6

if the user enters 5, you should generate:

11	18	25	2	9
10	12	19	21	3
4	6	13	20	22
23	5	7	14	16
17	24	1	8	15