Exercise 1:

Create and initialise two two-dimensional arrays with random numbers in the range 1 – 100. Each array is to have 10 rows and 5 columns. Print each array to the screen (neatly).

You must also add the two number matrices (arrays) together and display the result: For example:

Array 1				
47	92	38	76	33
11	33	83	54	12
4	61	73	83	94
58	3	13	63	11
47	79	55	1	49
3	1	89	70	40
55	52	25	31	99
61	78	46	6	88
84	67	95	81	10
58	80	48	52	79

Array 2				
86	72	34	41	87
76	75	36	88	76
91	76	87	9	79
71	9	66	17	93
56	82	7	38	36
29	68	53	77	7
90	25	98	96	44
15	62	7	63	52
54	9	75	49	13
56	79	16	35	25

Addition of Array1 and Array2				
133	164	72	116	120
87	108	120	142	88
95	137	160	92	173
129	12	80	80	104
103	161	61	39	84
32	69	143	147	47
145	77	123	127	143
76	140	52	69	140
138	76	170	131	23
114	159	64	87	104

Exercise 2:

Write a Java application to simulate the National Lottery's "Lotto" draw. You must implement the following:

1. **Set up data (45%):** Create the lotto test data. You must create 50,000 lotto tickets with randomly generated test data. A single lotto ticket contains two things.
   - A ticket number. Each ticket number must take the form '*xxabcd*' – where '*xx*' is a random number in the range 1 – 99 and '*abcd*' is a random selection of 4 letters from the alphabet. There is no requirement for each ticket number to be unique. The following are sample ticket numbers - 18FKIV, 94IGOG, 41LWNJ.
   - 6 unique numbers between 1- 45 inclusive.

2. **Display data (15%):** Display the generated test data neatly with each ticket number and its six numbers appearing on a separate line. The numbers for each ticket must be sorted in ascending order. The output will be something like the following:

```
Ticket 21NOGS: 1 4 19 37 39 43
Ticket 5CZAB: 5 8 11 12 14 26
Ticket 39PKGT: 7 12 14 15 21 42
Ticket 6DNJR: 4 9 25 29 36 45
Ticket 9LDVS: 2 13 18 25 26 31
Ticket 40PNUD: 15 31 32 38 44 45
```
etc…..

3. **Find winner (20%):** Display the ticket number(s) of any winning ticket(s) – for test purposes you can hard code the winning numbers if you wish and add them as a ticket to your collection of 500,000 tickets (thereby insuring a winner). It is possible that there will be no winner (unless of course you have planted one) and also that there may be more than one winner.

4. **Number analysis (20%):** Calculate and display the frequency with which each number in the lotto draw was "played" in the draw. This type of analysis might be of interest to statisticians/lotto players. The format of the output should be something like the following:

….

```
6 was drawn 65240 times
7 was drawn 65703 times
8 was drawn 65287 times
9 was drawn 65440 times
10 was drawn 65300 times
11 was drawn 65341 times
12 was drawn 65182 times
13 was drawn 65815 times
14 was drawn 65154 times
```
etc….

NOTE.

- Using methods will make your program easier to write/debug/test/read – you could create a method for each of the above four functions. In other words, you could use a method to set up the data, a method to display the data, a method to find the winner and a method to perform the number analysis. More methods could be added if necessary. There is a requirement that you must ***select one*** of the four functions above and ***implement it with a method***.

- The following method might prove useful. If passed a 2D int array it will sort each row in ascending order.

```
public static void sortEachTicket(int numbers[][]) {
    for (int[] row : numbers) {
        Arrays.sort(row);
    }
}
```

- If you need to sort a single dimensional array use code like the following.

```
Arrays.sort(anArray);
```

- The following piece of code might also prove useful as it allows you to initialise/reset an arrays contents to a particular value.

```
Arrays.fill(mostPayed, 0); //where mostPlayed is a single dimensional 1D int array
Arrays.fill(usedNumbers, false);//where usedNumbers is a single dimensional 1D boolean array
```

- All output to the screen is to be via System.out.println()

Other enhancements you could consider adding….

- Ensure that each generated ticket number is unique.

- Display the ticket numbers of any tickets that have matched 3, 4 and 5 numbers in the draw. Outputting also the number of tickets which have matched 3, 4 and 5.

- Instead of displaying the frequency of each number in the draw, display instead, the 6 numbers that are “played” the most from the test data generated.