Lab Exercise 7
Chapter 7 Arrays
Chapter 8 Multidimensional Arrays

COMP217 Java Programming
Spring 2019
Instructor: Gil-Jin Jang

Text: Liang, Introduction to Java Programming, Tenth Edition
Chapters 7 and 8

Ex7-1: Average of Random Numbers

- Objective: Write Java and C programs that
  1. Generates ***n*** random integers in [1 9] (including both 1 and 9) where ***n*** is given
  2. Print the generated numbers
  3. Compute the average of them
  4. Find out how many numbers are above the average
- Submission:
  - AverageRandom.java, AverageRandom.c

```
$ java AverageRandom
n?  9
9 0 6 3 2 5 7 2 8 1
Average = 4.3000
Number of values above
the average = 5
```

Ex7-2: Sorted Random Number Generator

- Objective: Write Java and C programs that
  - Generates ***n*** random integers in [1, 9], where ***n*** is given
  - Sort the generated numbers in an ascending order
  - Display them in the sorted order
  - When n <= 0, QUIT
- Submission:
  - SortedRandom.java, SortedRandom.c

```
$ java SortedRandom
n?  5
1 3 4 8 9
Input?  7
1 2 2 5 7 8 9
Input -1
```

(hint: keep the list sorted whenever a new number is generated)

Ex 7-3: Grading Multiple-Choice Test

Students' answer

	0	1	2	3	4	5	6	7	8	9
Student 0	A	B	A	C	C	D	E	E	A	D
Student 1	D	B	A	B	C	A	E	E	A	D
Student 2	E	D	D	A	C	B	E	E	A	D
Student 3	C	B	A	E	D	C	E	E	A	D
Student 4	A	B	D	C	C	D	E	E	A	D
Student 5	B	B	E	C	C	D	E	E	A	D
Student 6	B	B	A	C	C	D	E	E	A	D
Student 7	E	B	E	C	C	D	E	E	A	D

Key to the Questions:

	0	1	2	3	4	5	6	7	8	9
Key	D	B	D	C	C	D	A	E	A	D

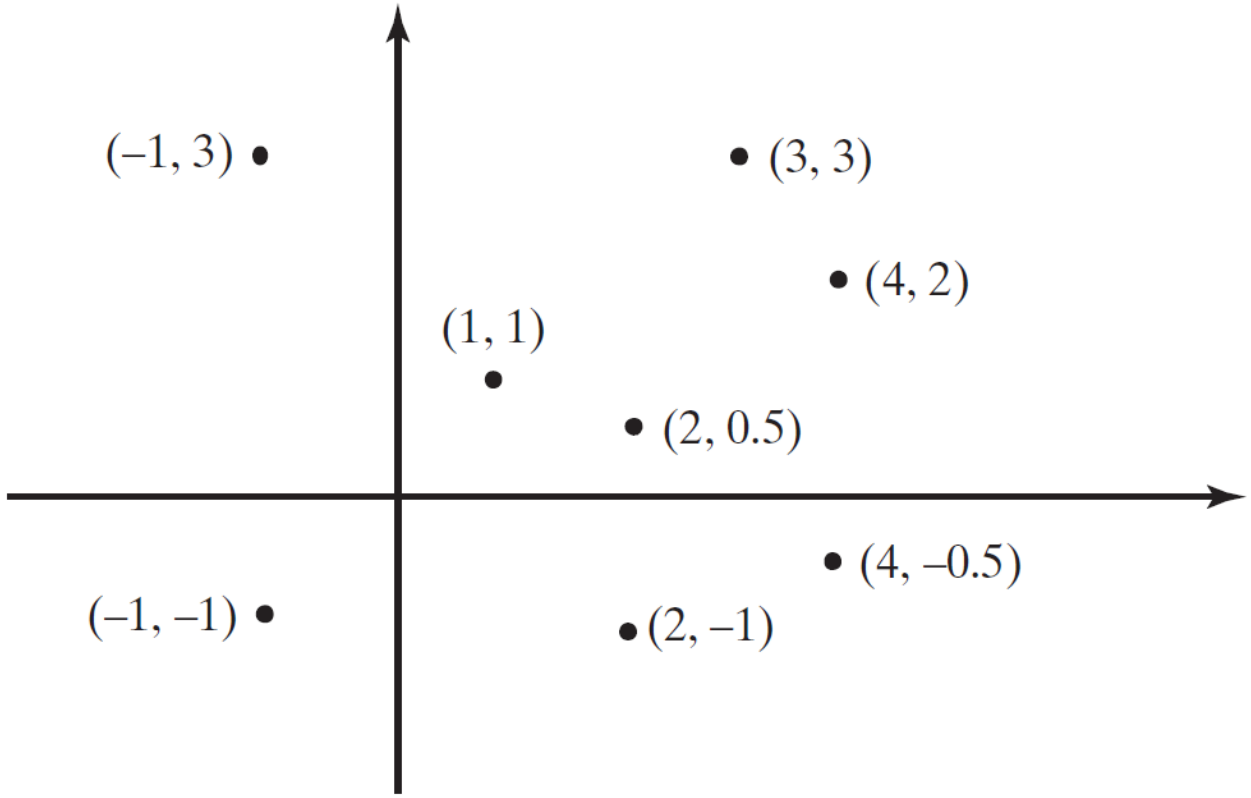
Write Java and C programs that grades multiple-choice test.

1. Use the answers and solutions in the example (no need to enter by Scanner class)
2. Display the number of correct answers of each student

Submission:

MultipleChoice.java, MultipleChoice.c

Ex 7-4: Finding Two Points Nearest to Each Other



	x	y
0	–1	3
1	–1	–1
2	1	1
3	2	0.5
4	2	–1
5	3	3
6	4	2
7	4	–0.5

Write Java and C programs that find two points that are nearest each other

1. Use the points in the example (no need to enter by Scanner class)
2. Display the points by: `(x1, y1) (x2, y2)`

Submission: FindNearest.java, FindNearest.c

Ex 7-5: Random Permutation

- Random permutation is reordering a list in a random order. The easiest way is to generate a number (order), and put an item in that position. Random shuffling in Dec of Cards example:

```
for (int i = 0; i < deck.length; i++) {
    int index = (int)(Math.random() * deck.length);
    int temp = deck[i];
    deck[i] = deck[index];
    deck[index] = temp;
}
```

- Another way of doing it is, keep a binary array as many as the length of the list, and set the array element as true if generated. If the array element is already true, generate another one
  - `boolean isGenerated[];`
  - `...`
  - `n = sc.nextInt();`
  - …
  - `isGenerated = new boolean[n]`

- Objective: Write Java and C programs that
  1. Generates a random permutation of 1…***n*** where ***n*** is given
  2. Your program should use a boolean (or int in C) array to check if the number is generated or not
  3. All the arrays should by dynamically-sized (use “new” in Java, and “malloc” in C)
- Submission:
  - RandPerm.java, RandPerm.c

```
$ java RandPerm
n?  9
9 1 6 3 2 5 7 8 4
```