SE 115 Introduction to Programming I

Lab No:	06
Topic:	Arrays

Scenario 0:

Consider the following array:

```
int[] myArr = {4, 8, 15, 16, 23, 42};
```

1. Answer the following questions without writing or running any code:
   a. What is the length of `myArr`?
   b. What is the value of the first element?
   c. What is the index value of the last element?
   d. What is the value of the third element?
   e. What is the value of `myArr[3]`?
   f. If we try to access `myArr[10]`, what happens?
2. Write a program that prints labeled answers for items (a)–(e). Then deliberately try to access `myArr[10]`, run the program, and explain the error message in a comment in your code.

Scenario 1:

SE115's current roster is stored in the school system as an integer array of length 20, where each entry is a student ID. After the enrollment process, you learn that the class size will increase. Write a program that defines a method which takes the old roster array and the desired new roster size as parameters. The method should return a new integer array of the given size. The first positions of the new array must contain the original IDs in the same order, and the remaining positions should be left empty (default values).

In `main`, create a sample old roster with student IDs starting at 1000 and increasing by 1. Call the method with a larger new size (for example, 42), and print both the old and new arrays to verify that the copying worked correctly.

Scenario 2:

Write a program that contains a function that rotates an integer array to the right by one position in a circular way (the last element becomes the first). The function should take the array as a parameter and return the rotated array. In the main method, generate a random integer array of size between 10 and 20, fill it with random values between 0 and 99, print the original array, call the function, and then print the resulting rotated array.

Scenario 3:

Using the `Random` class, write a program that simulates 1,000,000 independent days of weather. Each day should be one of four types with these target probabilities: Sunny 40%, Cloudy 30%, Rainy 20%, Stormy 10%. Create an array that represents these probabilities by including each category in the correct proportions, then for each simulated day pick a random position in that array to determine the day's weather and update a running total for each category. When the simulation finishes, print how many times each weather type occurred and its percentage (to two decimals) so you can compare to 40%, 30%, 20%, and 10%.

Scenario 4:

Write a program that creates an integer array of size 30 whose elements are random integers between 1 and 100. Consider the following definition:

- An index i is a "*peak*" if `arr[i]` is strictly greater than its immediate neighbors `arr[i-1]` and `arr[i+1]`; the first and last elements are not considered peaks.

Generate the array and count how many peaks there are. Print their values, the total number of peaks and the value of the tallest peak.

Bonus Question:

Write a program that creates a 2D integer array *scores* (5 students, 4 quizzes), fills it with random values in the range 0–100, and prints the table as a neat grid (each row on its own line). After printing, compute and print each student's average (row averages) and each quiz's average (column averages), formatted to two decimals. Finally, find the highest score in the table and print its value and its location as (`studentIndex, quizIndex`); if more than one cell shares the highest value, report the one with the smallest `studentIndex`, and if still tied, the smallest `quizIndex`.