

ONE-DIMENSIONAL and TWO-DIMENSIONAL ARRAYS

1. Write a program that reads student scores, gets the best score, and then assigns grades based on the following scheme:

Grade is A if score is \geq best - 10;
Grade is B if score is \geq best - 20;
Grade is C if score is \geq best - 30;
Grade is D if score is \geq best - 40;
Grade is F otherwise.

The program prompts the user to enter the total number of students, then prompts the user to enter all of the scores, and concludes by displaying the grades. Here is a sample run:

```
Enter the number of students: 4          ENTER
Enter 4 scores: 40 55 70 58             ENTER
Student 0 score is 40 and grade is C
Student 1 score is 55 and grade is B
Student 2 score is 70 and grade is A
Student 3 score is 58 and grade is B
```

2. Write a method that finds the three largest elements in an array of doubles using the following header:

```
public static double[] min(double[] array)
```

Write a method that takes an array of doubles and puts the array in order from smallest to largest with the following header:

```
public static void putInOrder(double[] array)
```

Write a test program that prompts the user to enter ten numbers, invokes the `min` method to return an array of the three largest elements. Then, invoke the `putInOrder` method to sort them from smallest to largest. Lastly, print out the three largest elements in order from smallest to largest. Here is a sample run of the program:

```
Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4.7 5 2          ENTER
The three largest numbers are: 4.7, 5, 6
```

3. Write a method that finds the product of two matrices of integers using the following header:

```
public static int[][] multiplyMatrices(int[][] m, int[][] n)
```

Write a method that returns whether two matrices of integers can be multiplied together using the following header:

```
public static boolean canBeMultiplied(int[][] m, int[][] n)
```

If `canBeMultiplied` returns false, then `multiplyMatrices` should return a double array with only one element of 0 in it. Two matrices can be multiplied if the number of columns of the first is equal to the number of rows of the second.

Example:
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1(1) + 2(2) + 3(3) \\ 4(1) + 5(2) + 6(3) \end{bmatrix} = \begin{bmatrix} 14 \\ 32 \end{bmatrix}$$

4. Write a method that returns the value of the largest positive element in a 2-D array, or 0 if all its elements are negative:

```
// Returns the value of the largest positive element in  
// the 2-D array m, or 0, if all its elements are negative.  
private static double positiveMax(double[][] m)
```

5. Write a method that determines whether a given number is a median for values stored in an array:

```
// Returns true if m is a median for values in the array  
// sample, false otherwise. (Here we call m a median if  
// the number of elements that are greater than m is the  
// same as the number of elements that are less than m)  
public boolean isMedian(double[] sample, double m)
```

Use one "for each" loop to traverse sample.

Write a test program and create multiple arrays in order and test your method several times. Verify that is it correct.