

Do not use ArrayList

Do not use enhanced for loop/for each loop

Do not write OOP

- (5 points) Enhance the **ArrayProcessing** program we completed in the lab.
 - Create an array of int to store 100 integers.
 - In the **printArray** method, print 10 elements per line.
 - Write a method that takes as its parameter as **int** array, and calculates and returns the standard deviation of the elements in the array as a double.

$$\text{Standard Deviation} = \sqrt{\frac{(x_1 - \text{mean})^2 + (x_2 - \text{mean})^2 + \dots + (x_1 - \text{mean})^2 + \dots + (x_n - \text{mean})^2}{n}}$$

Then in main method, call these user-defined methods with appropriate parameters and output the results.

Sample run:

```
Elements in an array of 100, organized by 10:
348 241 462 460 77 397 137 308 370 43
4 289 447 116 467 385 147 262 317 209
462 285 37 68 216 37 160 275 8 425
44 180 113 490 375 119 453 390 134 402
315 31 301 236 419 268 198 232 126 278
150 120 114 145 16 182 69 429 328 280
254 157 374 224 226 54 481 171 398 389
255 186 393 81 329 35 264 135 393 170
53 104 179 108 254 176 424 293 468 453
424 222 244 240 395 110 133 480 17 430

Mean: 245.72
Standard Deviation: 140.84
```

- (15 points) Download a data file (`City_Precipitation_Data.txt`). The data file contains the precipitation data for a number of selected US cities.

Write a program that uses **two arrays**, one to store the city names, and another to store the precipitations. The program also uses 4 methods to process the arrays.

- The 1st method** has two parameters (name array and precipitation array), it prints each city's name and precipitation
- The 2nd method** has one parameter (precipitation array), it finds and returns the index of the most precipitation.
- The 3rd method** has one parameter (precipitation array), it finds and returns the index of the least precipitation.
- The 4th method** has one parameter (precipitation array), it calculates and returns the average precipitation of all the cities.

In the main method, create two arrays and read data from the data file into the arrays, and call the methods as follows:

- Call the **1st method** to print each city's name and precipitation
- Call the **2nd method** to get the index of the highest precipitation, then print the city name with the highest precipitation (assume there is no tie).

- Call the **3rd method** to get the index of the least precipitation, then print the city name with the least precipitation (assume there is no tie).
- Call the **4th method** to get the average precipitation and print it.
- Then the program prints all the cities that have less than the average precipitation and all the cities that have more than or equal to the average precipitation.

Format the output appropriately.

Sample run:

Name	Precipitation in Inches
Billings	14.77
Honolulu	18.29
Houston	47.84
Indianapolis	40.95
Jackson	55.95
Jacksonville	52.34
Juneau	58.33
Kansas_City	37.98
Richmond	43.91
Knoxville	48.22
Birmingham	53.99
Bismarck	16.84
Bridgeport	44.15
Burlington	36.05
Albuquerque	9.47
Anchorage	16.08
Asheville	47.07
Charleston	51.53
Charleston	44.05
Charlotte	43.51
Atlantic_City	40.59
Portland	45.83
Portland	37.07
New_Orleans	64.16
New_York	49.69
Oklahoma_City	35.85
Sioux_Falls	24.69
Baltimore	41.94
Baton_Rouge	63.08
Columbia	48.27
Cheyenne	15.45
Cleveland	38.71
Columbus	38.52
Dallas-Ft_Worth	34.73
Des_Moines	34.72
Fairbanks	10.34
Grand_Junction	8.99
Grand_Rapids	37.13
Hartford	46.16
Las_Vegas	4.49
Lexington	45.91
Little_Rock	50.93
Washington	39.35
Long_Beach	12.94
Los_Angeles	13.15
Louisville	44.54
Milwaukee	34.81
Minneapolis	29.41
Montgomery	54.77
Mt._Washington	101.91
Nashville	48.11
Springfield	35.56
St_Louis	38.75
Philadelphia	42.05
Providence	46.45
Sacramento	17.93
Salt_Lake_City	16.50
San_Antonio	32.92
San_Diego	10.77
San_Francisco	20.11

Savannah	49.58
Seattle-Tacoma	37.07
Pittsburgh	37.85
Vero_Beach	51.93
Wilmington	42.81
Dodge_City	22.35
[66 cities]	

The city with the most precipitation is Mt._Washington: 101.91.

The city with the least precipitation is Las_Vegas: 4.49.

The average precipitation of all cities is 37.35.

The cities that have less than the average precipitation:

Billings	14.77
Honolulu	18.29
Bismarck	16.84
Burlington	36.05
Albuquerque	9.47
Anchorage	16.08
Portland	37.07
Oklahoma_City	35.85
Sioux_Falls	24.69
Cheyenne	15.45
Dallas-Ft_Worth	34.73
Des_Moines	34.72
Fairbanks	10.34
Grand_Junction	8.99
Grand_Rapids	37.13
Las_Vegas	4.49
Long_Beach	12.94
Los_Angeles	13.15
Milwaukee	34.81
Minneapolis	29.41
Springfield	35.56
Sacramento	17.93
Salt_Lake_City	16.50
San_Antonio	32.92
San_Diego	10.77
San_Francisco	20.11
Seattle-Tacoma	37.07
Dodge_City	22.35
[28 cities]	

The cities that have more than or equal to the average precipitation:

Houston	47.84
Indianapolis	40.95
Jackson	55.95
Jacksonville	52.34
Juneau	58.33
Kansas_City	37.98
Richmond	43.91
Knoxville	48.22
Birmingham	53.99
Bridgeport	44.15
Asheville	47.07
Charleston	51.53
Charleston	44.05
Charlotte	43.51
Atlantic_City	40.59
Portland	45.83
New_Orleans	64.16
New_York	49.69
Baltimore	41.94
Baton_Rouge	63.08
Columbia	48.27
Cleveland	38.71
Columbus	38.52
Hartford	46.16
Lexington	45.91
Little_Rock	50.93
Washington	39.35
Louisville	44.54

Montgomery	54.77
Mt. Washington	101.91
Nashville	48.11
St Louis	38.75
Philadelphia	42.05
Providence	46.45
Savannah	49.58
Pittsburgh	37.85
Vero Beach	51.93
Wilmington	42.81
[38 cities]	

Due: TBA

- To receive full credit, the assignment must be submitted to D2L by the due date.
- Late submissions will incur a penalty of 5% per day.

Style, form, documentation, naming convention, and more

Each program should have a file header section. /* * Author: Your name * Date: Date of completion * Assignment: Assignment # NameOfSourceCode.java * Description: The program description */	Up to 5% deduction
Each program should be written with the appropriate form and style. Use indentation, blank line, and comments to make the source code easy to read.	Up to 5% deduction
Use Java naming convention and meaningful names to name the classes, methods, variables, constants, and other identifiers in the programs.	Up to 5% deduction
Format the output appropriately	Up to 5% deduction