Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

20.3 Program 1c: World City Populations

Objective

- Print one and two dimensional arrays in very specific table format

Background reading

- Chapter 10.4 Formatting output. Read this first to learn about format specifiers.

Assignment

Did you know there are many cities in the world with population over 10 million people? Furthermore, none of the top twenty most populous cities are in the United States. The populations are such large numbers that they don't fit in an `int` type. We need to choose between a `long`--which makes sense because we have whole persons--or a `double`, which might be useful when we find percent. Don't worry. I made the choice for you, but you might challenge might choice with good cause!

You will write a program that displays city name, country, population, and percent of the country's population that live in the city. An example output follows (cities in all CAPS represent capitols).

```
City           Country          Population     Percent
_____________________________________________________
Shanghai       China            24,153,000       1.744
Seoul          South Korea      10,290,000      20.259
Chengdu        China            10,152,000       0.733
KINSHASA       Congo D.R.       10,125,000      11.732
LIMA           Peru              9,752,000      30.692
New York       United States     8,543,543      43.036
```

Data Structures

The country names can be stored in a one-dimensional String array. A corresponding two-dimensional array will hold the population counts: the city population and the country total population.

Organizing the program into methods

Your program must have these public static methods. Order them in your code where it makes sense to you, just make main() the first or last one.

- *main(String[] args)*
  - This method should consist only of calls to methods: getCities(); getPopulations(); printHeader(); printStats().
  - Two or four lines should be sufficient for the body of the method.
  - There are no loops in the main() method.
- *void printHeader()*
  - outputs table column header: City, Country, Population, Percent separated by spaces (no tabs!)
  - "Percent" means what percentage of the total country population lives in this city

  You will be tempted to do this:
  System.out.println(" City Country Population Percent");

  Let's not. Adjusting column width is much better in the long run if done using formatting methods. For each string, use %s as a placeholder for the string argument that follows. They will be substituted from left to right in the order given. To separate columns, add the width value and if you want it left-justified, add a minus sign.
  System.out.printf("%-15s%-17s%15s%12s\n", "FirstStringArg", "SecondStringArg", "ThirdStringArg", "FourthStringArg");
  Now, that is not precisely the correct widths, but what fun would it be if you couldn't experiment?
- *String repeat(char ch, int numOfCharsInString)*
  - Returns a string the length of numOfCharsInString consisting of only the character ch.
  - Instead of printing a dash (that is, the minus sign), let's print a Unicode character for the "em dash" which is a longer dash. Look up its value in a Unicode Table. It will show U+aNumber. For example, a space is U+0020. So get the number for the em dash and precede it with a backslash-u, for example, a space would be:
    char space = '\u0020';
- *void printStats(String[] cities, double[][] population)*
  - Loops through the parallel arrays printing data and performing the division calculation to produce the Percent value.
  - This time use System.out.format which is identical to System.out.printf but we are trying to build our language skills.
  - To print floating-point values, we use %f (if you mistakenly try to use %d it will expect a *d*ecimal number, that is, a whole number like int, short, etc.).
  - Here's a nifty trick: instead of outputting 24153000 or writing a dastardly method to insert commas, the method can insert them for us (Yay!). For example, %,12.0fwill print a minimum width of 12, no trailing decimal places, and insert commas between triples.
- *String getIdentificationString()*
  - Returns "Program 1c, Student Name"
- *double[][] getPopulations()*
  - Creates and returns a two-dimensional array containing the following data:

```
24153000,1384688986
18590000,1384688986
18000000,207862518
14657000,81257239
14543000,162951560
13617000,126168156
13197596,143964513
12877000,105920222
12784000,1384688986
12400000,1296834042
12038000,207652865
11908000,1384688986
11548000,1384688986
11035000,1296834042
10608000,1384688986
10355000,207862518
10290000,50791919
10152000,1384688986
10125000,86300000
9752000,31773839
```

- *String[] getCities()*
  - Creates and returns a one-dimensional String array containing the following data:

```
"Shanghai,China"
"BEIJING,China"
"Karachi,Pakistan"
"Istanbul,Turkey"
"DHAKA,Bangladesh"
"TOKYO,Japan"
"MOSCOW,Russia"
"MANILA,Philippines"
"Tianjin,China"
"Mumbai,India"
"Sao Paulo,Brazil"
"Shenzhen,China"
"Guangzhou,China"
"DELHI,India"
"Wuhan,China"
"Lahore,Pakistan"
"Seoul,South Korea"
"Chengdu,China"
"KINSHASA,Congo D.R."
"LIMA,Peru"
```

Notes:

- The first city+country name corresponds to the first row in the population array, and so on.
- Print only cities with population over 10 million. When traversing an array, use the array length property to ensure your program does not get an ArrayIndexOutOfBoundsException error.

LAB ACTIVITY | 20.3.1: Program 1c: World City Populations | 15 / 15

Submission Instructions

Compile command

```
javac Populations.java -Xlint:all -encoding utf-8
```

We will use this command to compile your code

Upload your files below by dragging and dropping into the area or choosing a file on your hard drive.

Populat...s.java Drag file here or **Choose on hard drive.**

Submit for grading

Latest submission - 11:08 PM on 09/08/19 — Submission passed all tests ✓ — Total score: 15 / 15

☐ Only show failing tests **Download this submission**

1: Compare output ^ 7 / 7

Your output

```
City            Country          Population
-------------------------------------------
Shanghai        China            24,153,000
BEIJING         China            18,590,000
Karachi         Pakistan         18,000,000
Istanbul        Turkey           14,657,000
DHAKA           Bangladesh       14,543,000
TOKYO           Japan            13,617,000
MOSCOW          Russia           13,197,596
MANILA          Philippines      12,877,000
Tianjin         China            12,784,000
Mumbai          India            12,400,000
Sao Paulo       Brazil           12,038,000
```

```
Shenzhen        China           11,908,000
Guangzhou       China           11,548,000
DELHI           India           11,035,000
Wuhan           China           10,608,000
Lahore          Pakistan        10,355,000
Seoul           South Korea     10,290,000
Chengdu         China           10,152,000
KINSHASA        Congo D.R.      10,125,000
```

2: Unit test 2 / 2

Test that the last entry in the populations array returned from `getCities()` is LIMA, Peru.

Test feedback

```
getCities() correctly returned "LIMA,Peru" a
```

3: Test repeat() 2 / 2

Show that repeat('%',91) returns a string of '%' chars with length 91. Show that repeat('\t',1) returns a string made up of a single tab character.

Test feedback

```
repeat() correctly returned
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

repeat() correctly returned "	"
```

4: Unit test 2 / 2

Test getPopulations() returns a 20x2 array.

Test feedback

```
getPopulations() has correct format
```

5: Unit test 2 / 2

Test getIdentificationString() returns Program 1, Student Name

Test feedback

```
getIdentificationString() correctly returned
```

5 previous submissions

11:07 PM on 9/8/19	8 / 15	View
11:07 PM on 9/8/19	8 / 15	View
11:05 PM on 9/8/19	6 / 15	View
11:01 PM on 9/8/19	6 / 15	View
10:48 PM on 9/8/19	8 / 15	View

Trouble with lab?