

**King Abdulaziz University**  
**Faculty of Computing and Information Technology**  
**Semester: Spring 2021**

|  |  |
| --- | --- |
| **Course Code: CPCS 203** | **Course Name: Programming II** |

**Assignment #1 (Saudi Arabia Water Resource Usage App)**

|  |  |
| --- | --- |
| **Assigned Date** | **09/02/2021** |
| **Delivery Date and Time** | **27/02/2021 @ 11:00 PM** |

**WARNING**:

* This program must ONLY be submitted on the Blackboard!
* This project worth 10% of the overall module marks (100%).
* NO assignment will be accepted after 11:59 pm for any reasons.
* Student can submit their assignments between 11 and 11:59 PM but in this case it will be consider as late submission, and they will lose 2 points from the total mark of the assignment.
* For discussion schedule, check the teacher name, date and time on the blackboard. **Further information is provided in the course syllabus.**

**Objectives:**

* Learn how to use and implement multidimensional array.
* Performing procedure on array elements.
* Learn to use and implement String, File I/O (Reading/Writing from/to files).

**Description**

According to reports, the water usage rate of Saudi Arabia is twice the world average. To deal with this alarming situation, the Department of Statistics has collected the total water usage data from various cities in different regions for the recent years along with the population details during these years. You are required to develop an application that can use this data as input and provide insight details of the water usage from different perspective such as the per year percentage increase or decrease in water usage over time in a specific city, the per capita water usage, the lowest and highest water usage cities etc. All this information will be helpful to help in reducing the water usage per capita.

Table 1 shows the data of regions, cities, population and the annual total water usage in cubic meter (cb.m).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Region | City | Year | Population | Water Usage cb.m |
| Riyadh | Ar-Riyad | 2018 | 5516374 | 635495744 |
| 2019 | 4503744 | 638347384 |
| 2020 | 4564633 | 650736422 |
| Al-Kharj | 2018 | 415865 | 43593433 |
| 2019 | 418636 | 42957485 |
| 2020 | 425330 | 44326383 |
| Ad-Dawadmi | 2018 | 263847 | 28453343 |
| 2019 | 269038 | 29504583 |
| 2020 | 271947 | 30129347 |
| Makkah | Makkah-al-Mukarramah | 2018 | 2119393 | 533563432 |
| 2019 | 2124848 | 545564533 |
| 2020 | 2042047 | 535765756 |
| Jeddah | 2018 | 3516374 | 345675764 |
| 2019 | 3503744 | 356006874 |
| 2020 | 3564633 | 335834763 |
| Taif | 2018 | 527834 | 21565633 |
| 2019 | 525437 | 22300454 |
| 2020 | 543823 | 22506565 |
| Qassim | Buraidah | 2020 | 419383 | 23594575 |
| Unaizah | 2020 | 119383 | 9557786 |

Table 1 Water Usage Data

Year [3][عدد المدن ع حسب المنطقة خلاص سويتها ][عدد السنوات ع حسب المدينة لسى]

From the given Table 1, you need to make different arrays to store all the data used in the program. For example, you need a one-dimension array to store region. Each region contains one or more cities, so you need a two-dimension array to store the city. Similarly, each city has records of several previous years. The program stores the year, population, and water usage in different column. Therefore, you need 3 three-dimension arrays: (first one to store the year, second one to store the population, and third one to store the water usage).

**Basic Requirements**

* The program must read the data from a text file called “input.txt” that follows a specific pattern. If the file doesn’t exist, print a message to let the user know what happened. – see **Input File Pattern** section for more details.
* The program must generate a text file as the output called “output.txt” that contains the results of the commands written in the input file – see **Output File Pattern** section for more details.
* The program must load and deal with data dynamically. This means you should never write the data as hard codes. You should use loops to deal with the arrays.
* The program must follow the format of the output file as possible.

**Input File Pattern**

The input file called ‘input.txt’ is attached with this document. The pattern, which the input file follows, is explained in the following input file example:

3 //The first line in the input file: the number of the region stored in this file. In other words, the size of region array.

3 3 2 //The second line in the input file: the number of cities in each region. It is separated with a space. Ex: the first region has 3 cities, the second region has 3 cities, and the third region type has 2 cities.

In other words,

cities[0] = new String[3];

cities[1] = new String[3];

cities[2] = new String[2];

**add\_regions Riyadh Makkah Qassim** //The add\_regions command registers the names of the region. It takes a list of the region names separated by a space. Ex:

regions[0] = “Riyadh”;

regions[1] = “Makkah”;

regions[2] = “Qassim”;

**add\_cities Riyadh Ar-Riyad Al-Kharj Ad-Dawadmi** //The add\_cities command is used to registers the names of the cities. It takes the name of the region as the first argument, and then the list of the city names separated by a space. “Riyadh” region has three cities, so three names are defined (Ar-Riyad, Al-Kharj, Ad-Dawadmi). Note: you have to find the index of the region and the index of the city to set the names.

**add\_cities Makkah Makkah-al-Mukarramah Jeddah Taif** // “Makkah” region has three cities, so three names are defined in the command (Makkah-al-Mukarramah, Jeddah, Taif).

**add\_cities Qassim Buraidah Unaizah** // “Qassim” region has two cities, so two names are defined in the command (Buraidah, Unaizah).

**add\_populations Riyadh Ar-Riyad 3** //”add\_populations” command registers the population details of the city for a region. It takes the name of the region as the first argument, the name of the city as the second argument, the number of the years as the third argument. All these arguments are separated by a space. This command is the followed by list of details including year, population, and Water usage - all separated by a space.

Ex: For “Riyadh” region and “Ar-Riyad” city, there are 3 records containing the year, population and water usage. As mentioned earlier, you need 3 three-dimension arrays; the first one to store the year as a string, the second one to store the population as an int, and the third one to store water usage as long.

2018 5516374 635495744

2019 4503744 638347384

2020 4564633 650736422

**add\_populations Riyadh Al-Kharj 3** //In the “Riyadh” region and “Al-Kharj” city, 3 years of record is available, and the information is listed below.

2018 415865 43593433

2019 418636 42957485

2020 425330 44326383

**add\_populations Riyadh Ad-Dawadmi 3** //In the “Riyadh” region and “Ad-Dawadmi” city, 3 years of record is available, and the information is listed below.

2018 263847 28453343

2019 269038 29504583

2020 271947 30129347

**add\_populations Makkah Makkah-al-Mukarramah 3** //In the “Makkah” region and “Makkah-al-Mukarramah” city, 3 years of record is available, and the information is listed below.

2018 2119393 533563432

2019 2124848 545564533

2020 2042047 535765756

**add\_populations Makkah Jeddah 3** //In the “Makkah” region and “Jeddah” city, 3 years of record is available, and the information is listed below.

2018 3516374 345675764

2019 3503744 356006874

2020 3564633 335834763

**add\_populations Makkah Taif 3** //In the “Makkah” region and “Taif” city, 3 years of record is available, and the information is listed below.

2018 527834 21565633

2019 525437 22300454

2020 543823 22506565

**add\_populations Qassim Buraidah 1** //In the “Qassim” region and “Buraidah” city, only 1 year of record is available, and the information is listed below.

2020 419383 23594575

**add\_populations Qassim Unaizah 1** //In the “Qassim” region and “Unaizah” city, only 1 year of record is available, and the information is listed below.

2020 119383 3989786

**print\_result Riyadh Ar-Riyad** //”print\_result” is a command that prints details about the per capita water usage and the change in percentage with respect to the previous year. It takes the name of the region as the first argument, the name of the city as the second argument. All these arguments are separated with a space. Ex: prints the details of the city “Ar-Riyad” in the “Riyadh” region.

**Hints:**

* Percentage change of per capita = [(This Year – Last Year) / Last Year] x 100
* Do not calculate percentage change for the first year of available record and simply print “NA” (for Not Applicable)
* Per capita daily usage (liters) = Total water usage in liters/ (total population x 365)
* 1 cb.m = 1000 liters

**print\_result Riyadh Al-Kharj** //Prints the details of the “Al-Kharj” city in the “Riyadh” region.

**print\_result Riyadh Ad-Dawadmi** //Prints the details of the “Al-Dawadmi” city in the “Riyadh” region.

**print\_result Makkah Makkah-al-Mukarramah** //Prints the details of the “Makkah-al-Mukarramah” city in the “Makkah” region.

**print\_result Makkah Jeddah** //Prints the details of the “Jeddah” city in the “Makkah” region.

**print\_result Makkah Taif** //Prints the details of the “Taif” city in the “Makkah” region.

**print\_result Qassim Buraidah** //Prints the details of the “Buraidah” city in the “Qassim” region.

**print\_result Qassim Unaizah** //Prints the details of the “Unaizah” city in the “Qassim” region.

**find\_lowest\_usage** // “find\_lowest\_usage” is a command that finds and prints the record of lowest per capita water usage.

**find\_highest\_usage** // “find\_highest\_usage” is a command that finds and prints the record of highest per capita water usage.

**about\_developer** // "about\_developer” is a command that prints the information about the programmer who created this program (Your information).

**exit** //” exit” is a command that ends the program. After this command, the output file called “output.txt” must be generated in the same directory of the program. Also, the date and time of the generation should be printed in the output file.

|  |
| --- |
|  |

Figure 1. Input file (input.txt)

**Output File Pattern**

The output file should include all the results of the commands that have been read from the input file “input.txt”. The format of the output file is provided as separate file “output.txt”.

**Commands You Have to Implement**

As you see in the input and output files, there are some commands that you have to implement:

|  |  |
| --- | --- |
| Command #1 | |
| **Command** | **add\_ regions** |
| **Description** | It registers the names of the region. It takes a list of the region names separated by a space. |
| **Example** | add\_regions Riyadh Makkah Qassim |
| **Output of the Example** | [Command] add\_regions  + Riyadh + Makkah + Qassim |

|  |  |
| --- | --- |
| Command #2 | |
| **Command** | **add\_cities** |
| **Description** | It registers the names of the cities. It takes the name of the region as the first argument, and then the list of the cities separated by a space. |
| **Example** | add\_cities Riyadh Ar-Riyad Al-Kharj Ad-Dawadmi |
| **Output of the Example** | [Command] add\_cities  -> City: Riyadh  + Ar-Riyad + Al-Kharj + Ad-Dawadmi |

|  |  |
| --- | --- |
| Command #3 | |
| **Command** | **add\_populations** |
| **Description** | It registers the year, population, and water usage details of the city. It takes the name of the region as the first argument, the name of the city as the second argument, the number of year as the third argument. All these arguments are separated with a space. Then it lists the year, population and water usage separated with a space. |
| **Example** | add\_populations Riyadh Ar-Riyad 3  2018 5516374 635495744  2019 4503744 638347384  2020 4564633 650736422 |
| **Output of the Example** |  |

|  |  |
| --- | --- |
| Command #4 | |
| **Command** | **print\_result** |
| **Description** | It prints details about the year, population, annual water usage, per capita daily usage and change (in percentage) of the specific city in a region. It takes the name of the region as the first argument, the name of the city as the second argument. All these arguments are separated with a space. |
| **Example** | print\_result Riyadh Ar-Riyad |
| **Output of the Example** |  |

|  |  |
| --- | --- |
| Command #5 | |
| **Command** | **find\_lowest\_usage** |
| **Description** | It is a command that finds and prints the lowest water usage per capita record. |
| **Example** | find\_lowest\_mileage |
| **Output of the Example** |  |

|  |  |
| --- | --- |
| Command #6 | |
| **Command** | **find\_highest\_usage** |
| **Description** | It is a command that finds and prints the highest water usage per capita record. |
| **Example** | find\_highest\_mileage |
| **Output of the Example** |  |

|  |  |
| --- | --- |
| Command #7 | |
| **Command** | **about\_developer** |
| **Description** | It prints the information about the developer of the guide (student info). |
| **Example** | about\_developer |
| **Output of the Example** | - Command: about\_developer  -> Developed By: Ali Alqahtai  -> University ID: 2004832  -> Section: DA |

|  |  |
| --- | --- |
| Command #8 | |
| **Command** | **exit** |
| **Description** | It ends the program. After this command, the output file called “output.txt” must be generated in the same directory of the program. Also, the date and time of the generation should be printed in the output file. |
| **Example** | exit |
| **Output of the Example** | Thank You! :)  Report generated on Sun Feb 07 17:18:32 AST 2021 |

**Deliverable**

You have to submit only the java file of your code. The file and the class names should be “**Project\_1\_YourFirstName\_YourLastName\_YourID**”. Where “YourFirstName“ is your first name, “YourLastName“ is your last name, “YourID“ is your type ID.

**NOTE:** your name, ID, and section number must be included as comments in the file!

**Output Format**

Your program must generate output in a similar format to the given sample output file [output.txt].

**Good Luck and Start Early!**