

1 Instructions

You may work in pairs (that is, as a group of two) with a partner on this lab project if you wish or you may work alone. If you work with a partner, only one of you will submit the lab project to Canvas for grading. To ensure you each receive the same score, it is imperative that you follow the instructions in §4 *What to Submit for Grading and by When* (i.e., put both of your ASURITE ID's in the submission archive filename and write both of your names, ASURITE ID's, and email addresses in the AUTHOR1: and AUTHOR2: lines of the header comment block for each source code file). What to submit for grading, and by when, is discussed in §5; read it now.

2 Lab Objectives

After completing this assignment the student should be able to,

- Complete all of the objectives of the previous lab projects.
- Write a nested for loop.
- Write a nested while loop.
- Read real data and parse it.
- Use 1D and 2D arrays and learn its advantages and disadvantages.

3 Prelab Exercises

3.1 Prelab Exercise 1: Read Software Requirements

Read §4 *Software Requirements* of this lab project document which describes what the lab project program will do, i.e., the **software requirements**. Then come back here and complete the remaining prelab exercises.

3.2 Prelab Exercise 2: Get familiar with nested loops

You can read the following website: https://www.tutorialspoint.com/cplusplus/cpp_nested_loops.htm

3.3 Prelab Exercise 3: Get familiar with 1D and 2D arrays

You can use the following website: <http://www.cplusplus.com/doc/tutorial/arrays/>

3.4 Prelab Exercise 4: Pseudocode

The pseudocode is given in the comments of the main.cpp file

4 Lab Exercise

This project will read data from about energy consumption from the website data.gov, more specifically from: <https://catalog.data.gov/dataset/utility-energy-registry-monthly-zip-code-energy-use-beginning-2016>. We will be using the .csv file (comma separated values). To simplify this exercise, we are only going to be using the first 50 entries. The modified file is provided to you in the input folder.

Then, our program will read the information using nested while and for loops and store it in a 2D array. The output that will be displayed in the terminal is as follows:

The output is well-formatted, however when exporting to pdf is difficult to maintain the spacing. Please see the folder output in the zip folder.

Class	ZIP city	ZIP Code	Value
electricity	Cato	13033	0
electricity	Bronx	10471	9144.86
electricity	Medina	14479	0
electricity	Auburn	13021	0
natural_gas	Troy	12182	-999
electricity	Colton	12927	62
electricity	Cape Vincent	13618	261
electricity	Marathon	13803	0
electricity	Liverpool	13090	1489
natural_gas	Marcy	13403	60324
electricity	Clayville	13322	23
electricity	Rome	13441	0
electricity	Broadalbin	12025	-999

electricity	West Leyden	13489	0.13
electricity	Perrysburg	14129	-999
electricity	Schenectady	12302	36.993
electricity	Alexandria Bay	13607	1.302
natural_gas	Pulaski	13142	68738
natural_gas	Oneida	13421	850
electricity	Schenectady	12309	1419.968
electricity	Melrose	12121	120
electricity	Clayton	13641	-999
natural_gas	Rensselaer	12144	1231
electricity	Johnsburg	12811	38
natural_gas	Verona	13478	-999
electricity	Basom	14013	0.475
electricity	Syracuse	13210	45.349
electricity	Cicero	13039	21.547
electricity	Stockton	14784	25.016
electricity	Liverpool	13088	23.23
electricity	Chautauqua	14722	-999
electricity	Angola	14006	622.332
electricity	Wellsville	14707	-999
electricity	Manlius	13104	4682.414
natural_gas	Queensbury	12804	233
electricity	Rochester	14625	0
natural_gas	Wells	12190	-999
electricity	Hastings	13076	654.962
electricity	Colton	12927	0.09
electricity	Woodgate	13494	67
electricity	Buffalo	14222	1314.348
electricity	Youngstown	14174	403
electricity	Averill Park	12018	187
electricity	Waddington	13694	22
electricity	Burt	14028	312.838
electricity	West Monroe	13167	0.316
electricity	Alexander	14005	-999
natural_gas	Wells	12190	0
electricity	Jordanville	13361	17
natural_gas	Rock City Falls	12863	-999

The city with the most consumption was: Pulaski with an energy use of 68738 MWh.

Data retrieved using the first 50 entries from: <https://catalog.data.gov/dataset/utility-energy-registry-monthly-zip-code-energy-use-beginning-2016>

4.a) Display the Table

It will display a table with four columns and the first fifty entries.

4.b) City with most consumption

At the end we will iterate over the array looking for the city that had the largest energy consumption. Finally we will report the sources of the data.

4.1 Additional Programming Requirements

1. Modify the **header comment block** at the top of main.cpp source code file so it contains the first author's information in the AUTHOR1: comment. If there is only one author, then delete the AUTHOR2: comment. However, if there are two authors, document the second author's information in the AUTHOR2: comment.
2. Always write your code in a way to **enhance readability**. This includes **writing comments** to explain what the code is doing, **properly indenting** the code, and **using blank lines** to separate the various parts of the program. For guidance, study the example programs in the textbook and the source code the instructor writes in class or posts online.

5 What to Submit for Grading and By When

- Note: There will be a 40% project penalty deduction (or -2 points) for not following these instructions and uploading incorrect files and/or incorrectly named files.
- Using the file explorer program of your operating system, create a new empty folder named **lab09-asurite** where *asurite* is your ASURITE username that you use to log in to MyASU (e.g., mine is *kburger2* so my folder would be named **lab09-kburger2**). If you worked with a partner on the project, put both of your ASURITE usernames in the name of the new folder **lab09-asurite1-asurite2**, e.g., Fred Flintstone's and Barney Rubble's project submission archive would be named **lab09-fflint-brubble.zip**.
- Copy your completed *main.cpp* source code file to the folder you just created. This folder should contain only one file named *main.cpp*. Do **not** copy the input file *payroll.txt* or the output file *paycheck.txt* to this folder. We do not need the data files because we will use our own when testing your program.
- Compress the folder creating a **zip archive** named **lab09-asurite.zip** or **lab09-asurite1-asurite2.zip**.
- Submit the zip archive to Canvas by the deadline using the *Lab Project 9* submission link.
- If you worked with a partner, then only one partner of the team submits to Canvas, but you will each receive the same score as long as you follow the rules for documenting both authors in the header comment block and properly naming your zip archive.
- If your program does not compile or run correctly, upload what you have completed for grading anyway because you will generally receive some partial credit for effort.
- The submission deadline can be found in the Canvas Lab 9 submission page. It is **Sat 24 Apr 11:59pm**.
- The Canvas submission link will become unavailable exactly at 11:59:00pm on the Mon following the Sat deadline and you will not be able to submit your project for grading after that time. We do not accept emailed submissions.
- Consult the online syllabus for the late project submission and academic integrity policies.

6 Grading Rubric

This section will be completed later.

6.1 Assign Bonus/Penalty Points. Deadline was 11:59pm Sat 24 Apr

- a. Deduct 0.5 pt (-10%) for a submission on Sun 25 Apr.
- b. Deduct 1 pt (-20%) for a submission on Mon 26 Apr.
- c. After Mon 26 Apr, late projects are not accepted for any reason.