Guidebook

Created By

Golam Morshed

TABLE OF CONTENTS

Description of the Program **1**

Node Installation Process **2**

Running the Anagram Grouping Program **4**

Result **5**

Source Code **6**

Links **7**

TABLE OF FIGURES

Figure 1: Run Code on Terminal Example (Mac OS) _____ 4

Figure 2: Output Result _____ 5

Figure 3: Source Code _____ 6

DESCRIPTION OF THE PROGRAM

The purpose of this program is to receive a list of words from a file, sort the words according to their anagrams, and then output the results to the terminal in the form of a table. An anagram is a word like "listen" and "silent" that is created by rearranging the letters of another word. One word per line should be present in the file that the programme first reads from. After that, it arranges each word's characters to provide a special key for recognising anagrams. Words that have the same characters arranged together are grouped. Following the anagram grouping, the program uses the cli-table package to display the results in a table style, making it simple to see which words are anagrams of one another. You must have Node.js installed on your computer to use this application. To execute the program from the terminal, save the program code in a file, prepare a text file containing the list of terms, and pass the filename as an argument. The output will appear on the terminal as a well-formatted table with the grouped anagrams displayed in it.

NODE INSTALLATION PROCESS

Installation on Windows

1. **Download the Installer:**
    - Go to the official Node.js website.
    - Click on the "Download" button to download the Windows installer (an .msi file).

2. **Run the Installer:**
    - Locate the downloaded .msi file and double-click it to start the installation.
    - Follow the prompts in the installer:
        - Accept the license agreement.
        - Choose the installation location (or use the default).
        - Ensure the "Add to PATH" option is selected.

3. **Verify the Installation:**
    - Open the Command Prompt (cmd) or PowerShell.
    - Type node -v and press Enter to check the Node.js version.
    - Type npm -v and press Enter to check the npm version.

4. **Install cli-table:**
    - In the Command Prompt or PowerShell, navigate to your project directory.
    - Run npm install cli-table to install the cli-table package needed for the program.

Installation on macOS

1. **Download the Installer:**
    - Navigate to official Node.js website.
    - Click on the "Download" button to download the macOS installer (a .pkg file).
2. **Run the Installer:**
    - Locate the downloaded .pkg file and double-click it to start the installation.
    - Follow the prompts in the installer to complete the installation.
3. **Verify the Installation:**
    - Open the Terminal.
    - Type node -v and press Enter to check the Node.js version.
    - Type npm -v and press Enter to check the npm version.
4. **Install cli-table:**
    - In the Terminal, navigate to your project directory.
    - Run npm install cli-table to install the cli-table package needed for the program.

RUNNING THE ANAGRAM GROUPING PROGRAM

1. **Prepare a word list file**
    - Create a text file containing your words, with one word per line (e.g., words.txt).

2. **Run the Program:**
    - Open the Command Prompt (Windows) or Terminal (macOS).
    - Navigate to the directory where your program and word list file are located.
    - Run the program using node your_program_file.js words.txt, replacing your_program_file.js with the actual name of your JavaScript file **(group_anagrams.js)** and words.txt with your word list file.

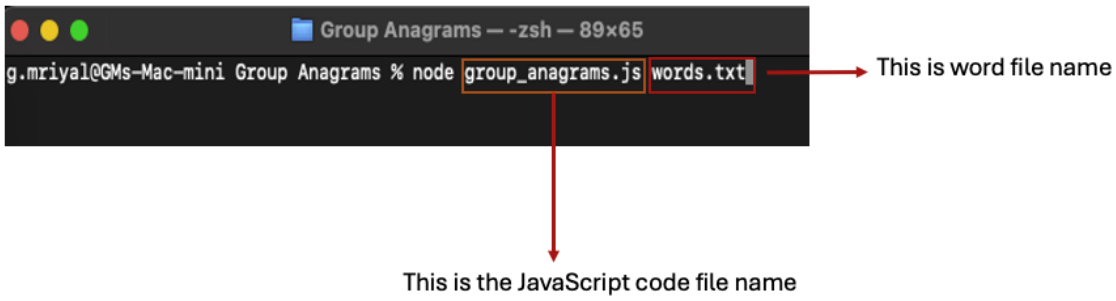

FIGURE 1: RUN CODE ON TERMINAL EXAMPLE (MACOS)

RESULT

Figure 2 show how the result will be output if we follow the above steps.

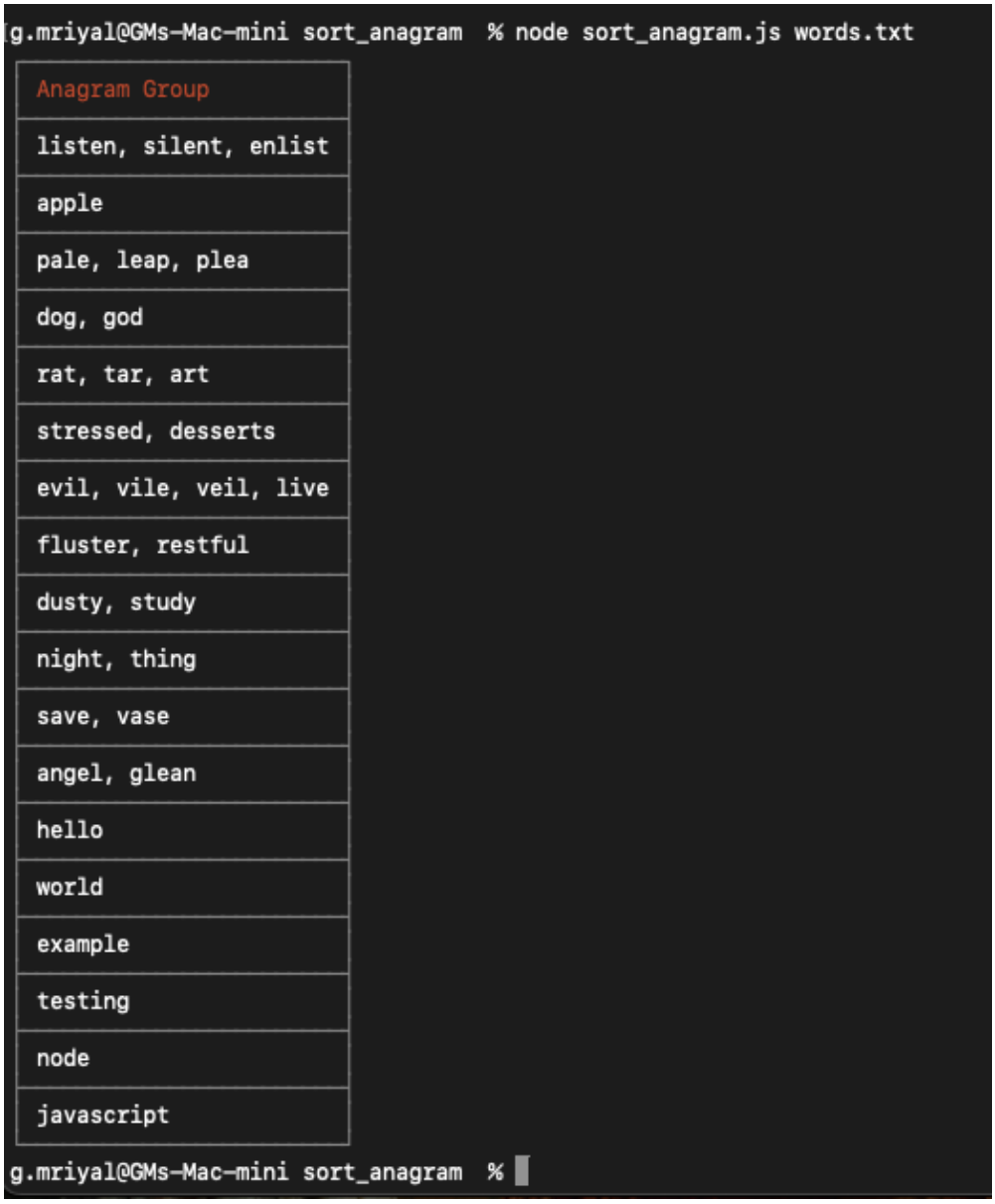

FIGURE 2: OUTPUT RESULT

SOURCE CODE

group_anagrams.js × words.txt

Users > g.mriyal > Desktop > Group Anagrams > group_anagrams.js > groupAnagrams

```
const fs = require('fs');
const Table = require('cli-table');

// Function to group anagrams
function groupAnagrams(words) {
    const anagramWordGroups = new Map();
    words.forEach(word => {
        // Split each word to array of its individual characters.
        // Use sort method to convert the characters alphabatically.
        // Use join method to join the characters to make a single string
        const sortedWord = word.split('').sort().join('');
        // Use the sorted word as a key for the anagram group
        if (!anagramWordGroups.has(sortedWord)) {
            anagramWordGroups.set(sortedWord, []);
        }
        anagramWordGroups.get(sortedWord).push(word);
    });

    return anagramWordGroups;
}

// Function to read file and process words
function processFile(filename) {
    fs.readFile(filename, 'utf8', (err, data) => {
        if (err) {
            console.error(`Error reading file: ${err}`);
            return;
        }
        const words = data.split('\n').map(word => word.trim()).filter(word => word.length > 0);
        const anagramWordGroups = groupAnagrams(words);
        displayAnagramGroups(anagramWordGroups);
    });
}

// Function to format and display anagram groups
function displayAnagramGroups(anagramWordGroups) {
    const table = new Table({
        head: ['Anagram Group']
    });

    anagramWordGroups.forEach(group => {
        table.push([group.join(', ')]);
    });

    console.log(table.toString());
}

// Get the filename from command line arguments
const filename = process.argv[2];
if (!filename) {
    console.error('Please provide a file to process.');
    process.exit(1);
}

// Process the provided file
processFile(filename);
```

FIGURE 3: SOURCE CODE

LINKS

GitHub:

https://github.com/GolamMorshed/group_anagrams

Download File if unable to clone from GitHub:

https://drive.google.com/file/d/1KpuYYaxTc9rjomaJOw943kP31AfWTewb/view?usp=sharing

Video tutorial to run the code:

https://drive.google.com/file/d/1u94wGPz-GSopb9xuuYY6IB9SDaTcCNOC/view?usp=sharing