**Carbon Footprint Calculation for a Household**

Authors: Khadiza Akter, Alonso Ibarra, Meng Huan Lee

Class: CIS-5-48643

Date: 12/09/21

This program can calculate the carbon footprint per year for a household. Three different main emission sectors, such as household usage energies, vehicles, and waste, were used to calculate household carbon emissions. Various emission calculation equations, standard values/constants, and program-related data were extracted from the environmental protection agency (EPA)’s excel file, which downloaded from the (<https://www3.epa.gov/carbon-footprint-calculator/data/GHGCalculator.xls>).

The emission factor is constant but depends on energy types, where natural gas, fuel oil, and propane emission were defined as constant for each fuel type in this program. However, the emission factor for electricity is varied according to the zip code. Therefore, a CSV file containing the zip code, state code, and electricity emission factor was considered as an input file and read all records from the CSV file, where a total of 41,255 rows for different zip codes. The **ReadCSVFile** function read all rows/records of the CSV file. Then electricity emission factor was searched using the user zip code. The **SearchZipCode** function was defined to find the electricity emission factor. As all types of energy follow the same equation to calculate emission, therefore, a user-defined function named **CalculatedEnergyEmission** was defined to calculate the energy-related emissions; therefore, it reduced the number of lines of coding significantly. Also, the program will offer the user to make some plans including sleep feature for computer and monitor, reduce usage of dryer, use energy saving bulbs, energy star model refrigerator for reducing household emission. User may take these plans or not. The general equation for these energy saving plans is **Energy Saving = yearly energy saving for taking a plan or action \* electricity emission factor**. The yearly energy saving depends on various plans and constant for each plan as well, and these are defined as constant in the program. The energy saving follows a general formula, therefore, a function **EnergyEmissionSaving** was defined to calculate all these above-mentioned saving plans.

The total number of vehicles for a household is a user input to determine the vehicle's emission. Then calculate the vehicle emission using user inputs, namely total driven mileages per year and mileages driven per gallon of each vehicle. A general formula to calculate the vehicle emission is **Vehicle emission = (Driven Miles Per Year / Mileages Per Gallon) \* Emission Factor for Passenger Vehicle \* Non-CO2 Vehicle Emission Ratio, w**here emission factor for passenger vehicle and non-CO2 vehicle emission ratio are constant values and declared as constants. A user-defined function named **CalculateVehicleEmission** was defined to calculate the vehicle emission for all household vehicles. Also, the program will offer the user to make plans to reduce the emission. Emission savings of a vehicle follow a similar equation as vehicle emission calculation. Therefore, we re-used the same **CalculateVehicleEmission**function to determine the vehicle emission savings. As a result, it reduced the total number of lines of coding significantly.

 Usually, waste emission was calculated from the number of people in a household and the average pound (lb) of emission as CO2 equivalent generated from waste per person per year. The equation for waste emission is **waste emission = number of people in household \* average pound (lb) CO2 equivalent generated from waste per person per year**. Next, the program asks the user whether they recycle different household waste materials (i.e., aluminum or steel cans, plastic, glass, newspaper, magazine) or not. If they recycle the materials mentioned above, it will reduce the waste emission. The emission reduction formula for recycling material is **waste emission = number of people in household \* recycle material avoid emission**. The term recycle material avoid emission depends on the type of materials, and it is a constant value that also is found in the EPA excel file. Also, this program offers waste emission reduction plans whether a household can start recycling for various materials. If they take any action, it will reduce the emission. The emission reduction formula is the same as the previous formula (**waste emission = number of people in household \* recycle material avoid emission**). To manage the waste emission, a single function named **CalculatedWasteEmission** was enough for multiplying two parameters which reduced total lines of code significantly.

Finally, a summary report for carbon footprint equivalence CO2 (lbs) per year of a household is displayed. The summary report shows the total emission for household energies, vehicles, and waste emissions before taking any saving action or plan. Then it shows the total emission savings from the household energies, vehicles, and waste emissions depending on the user plans. New total emission after taking the savings plans is also displayed—moreover, the total number of trees needed for absorbing the emission that is shown in the program. The summary report is also written in a text file, and the file name is **CarbonFootPrint.txt**. The function **WriteSummaryReport**performs the written operation for the summary report.

When a user runs the program, it will ask the number of people in a household and zip code as general information. Then it will ask for household energies information, vehicle information, and waste information.

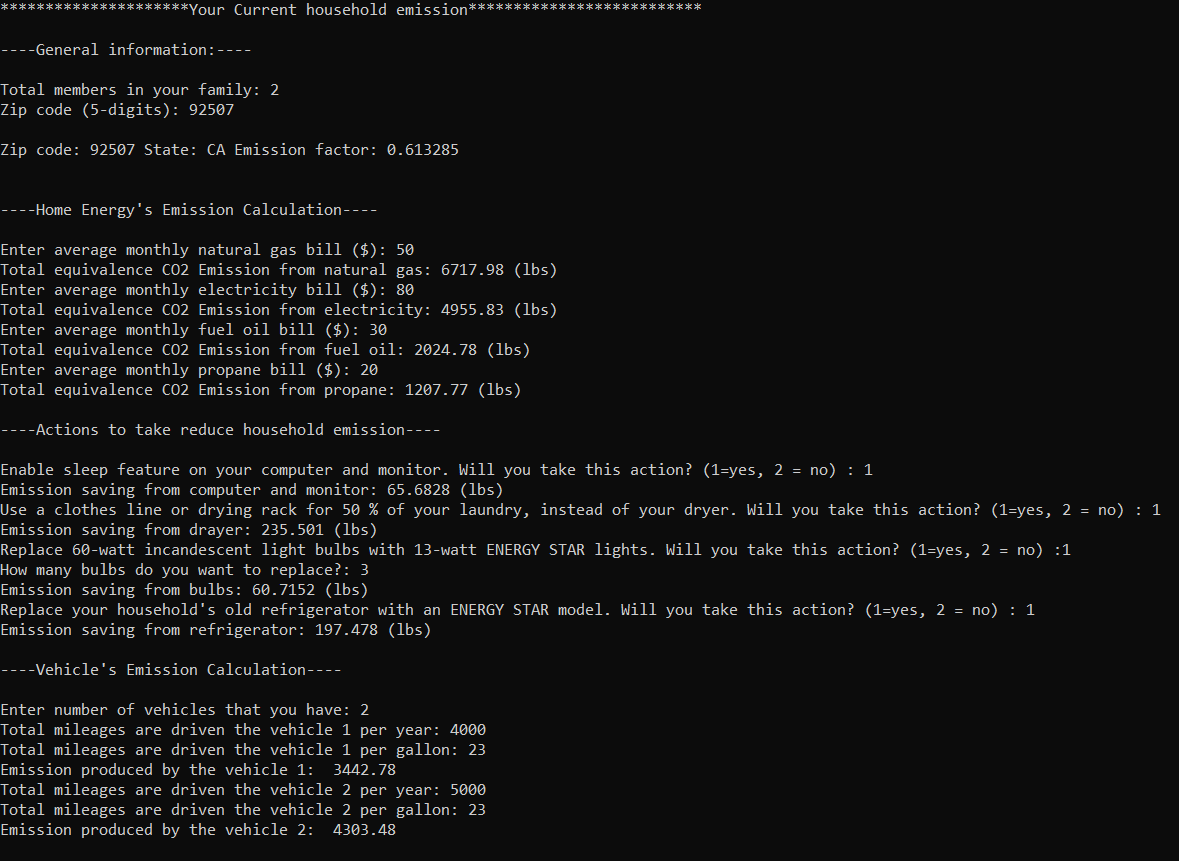
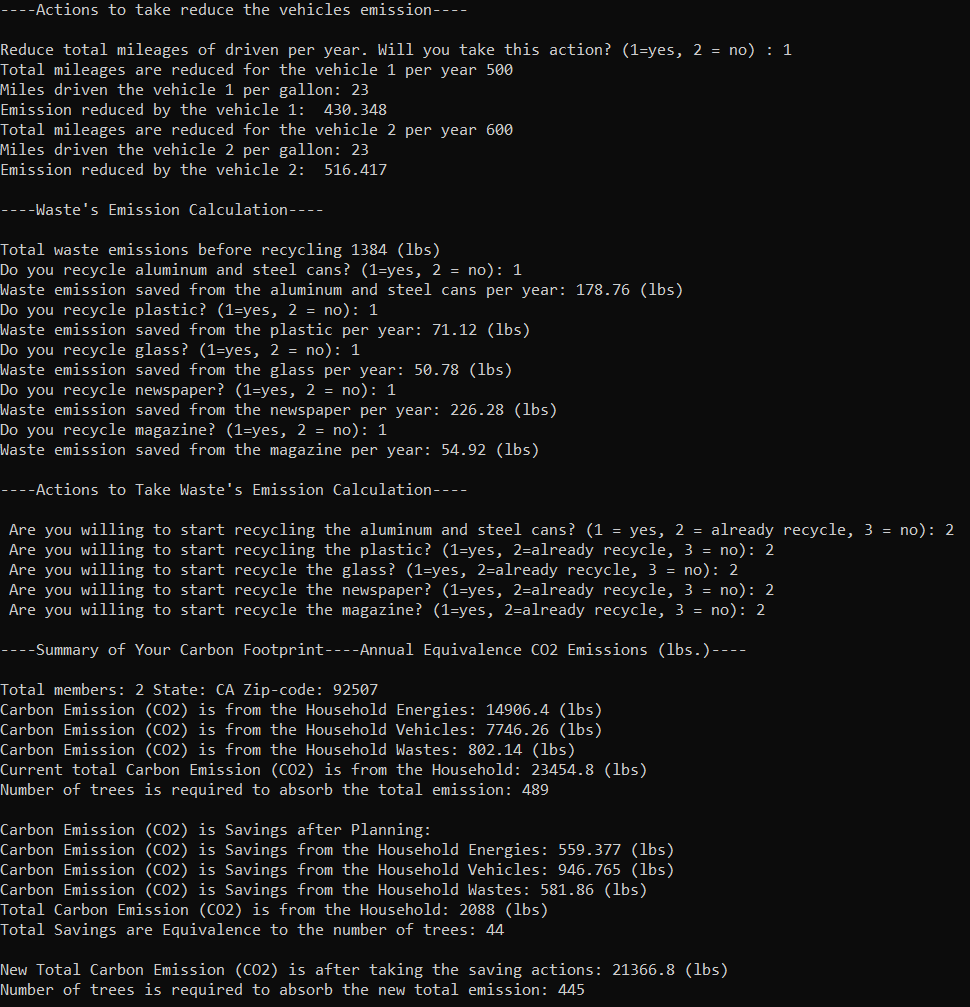
**Sample Screenshot of the Program:**

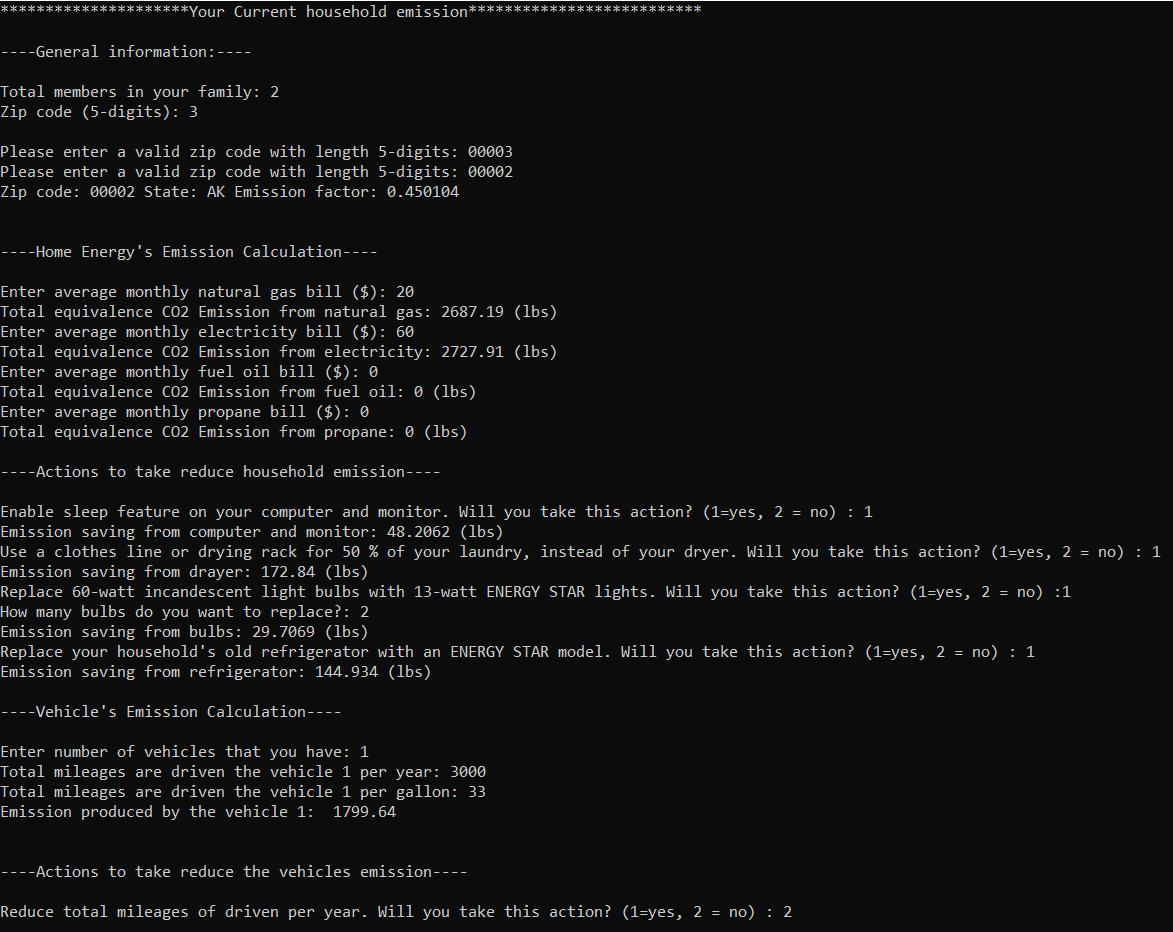
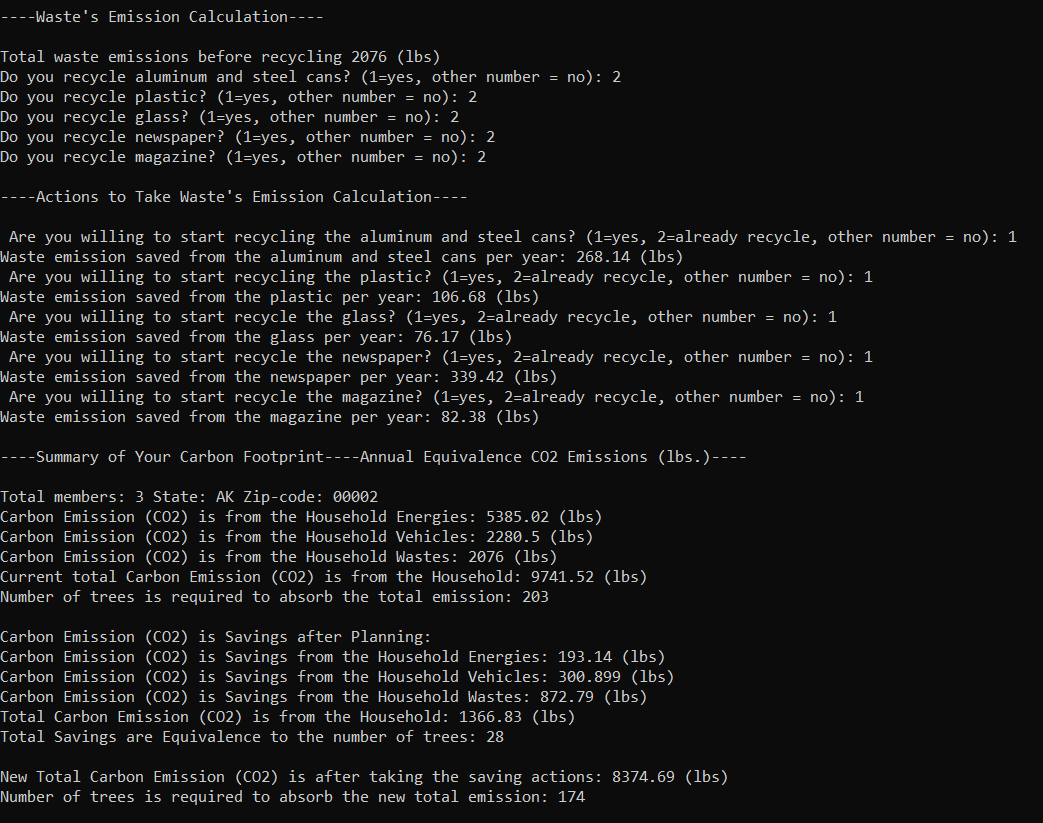
Screenshot #1 showed a general run of the program. The user inputs were the number of household members and zip code. The program displayed zip code, state code, and electricity emission factor if the zip code is found. In the next step, the user provided the various utility bills, and the program displayed the corresponding equivalence CO2 emission. Then program provided several options for reducing energy-related emissions, and the user agreed to take all the plans. The program showed total savings for each action. The next step is to calculate the vehicle emission, ask the total number of vehicles in the household and driven mileages per year and mileages per gallon. The program displayed the calculated vehicle emission. Then the program asks the user to take vehicle emission-related action. And program calculated the saving emissions from each vehicle. The next step is to calculate the waste emission-related emission. Total emission from the household waste was calculated from the number of household members as input and then asked whether the household person recycles any waste or not. The program showed the total amount of saved emission from any recycled materials. Then the program asks the user to take the recycling actions for different materials to reduce waste-related emissions. A summary report was displayed in the final part of the program.

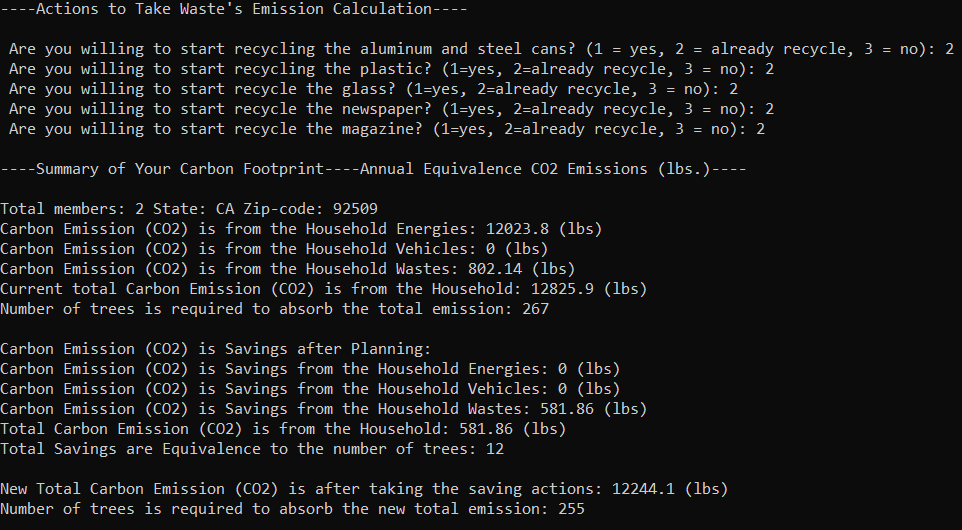
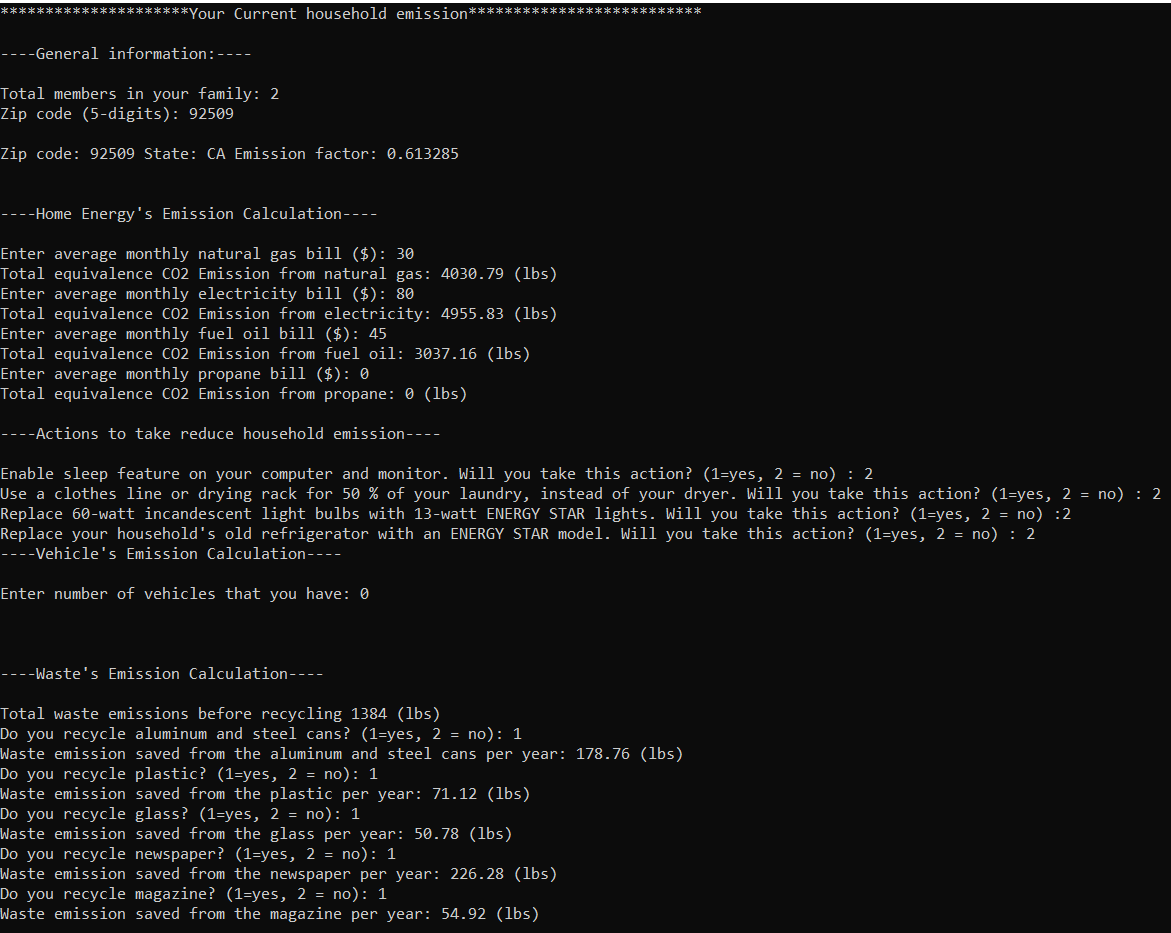
Screenshot # 2 showed that the user provided the wrong zip code and asked for a valid zip code with a length of 5 digits. Also, some utility bills were inputs as zero, and generated emission was zero as well. This time user-provided one vehicle information. In case of waste emission, the user did not recycle any material, then actions or plans user wanted to recycle all materials. The program showed the emission savings from each material. Also, a summary report of the carbon footprint was shown in screenshot # 2.

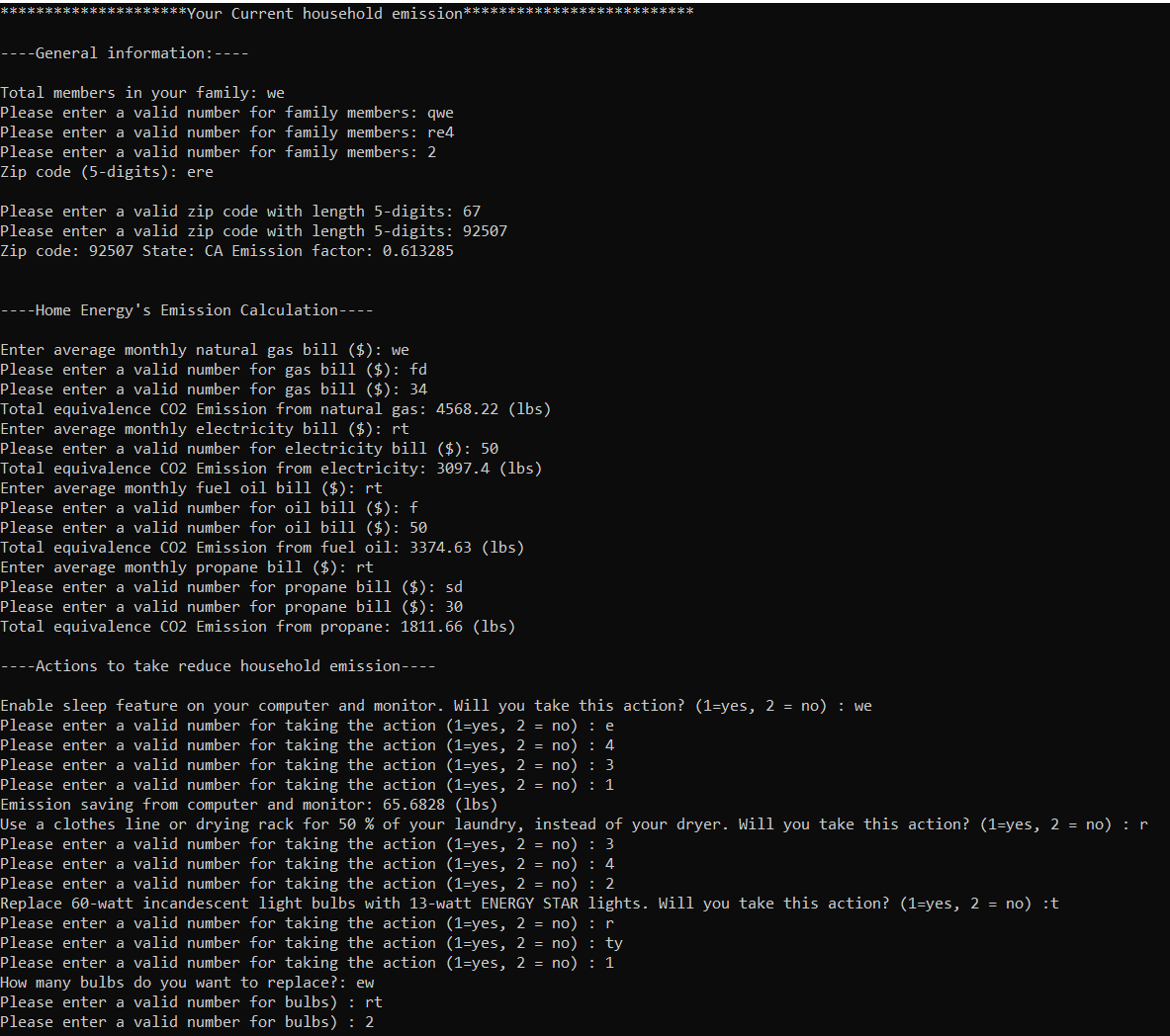
In screenshot # 3 user did not take any action to save the household energy emissions. Also, the user did not have a car. User recycled all materials and reduced the waste emission. Finally, the program showed the summary report of carbon footprint. The screenshot # 4 shows the invalid inputs for many cases and the program waits for a valid input with a meaningful message.

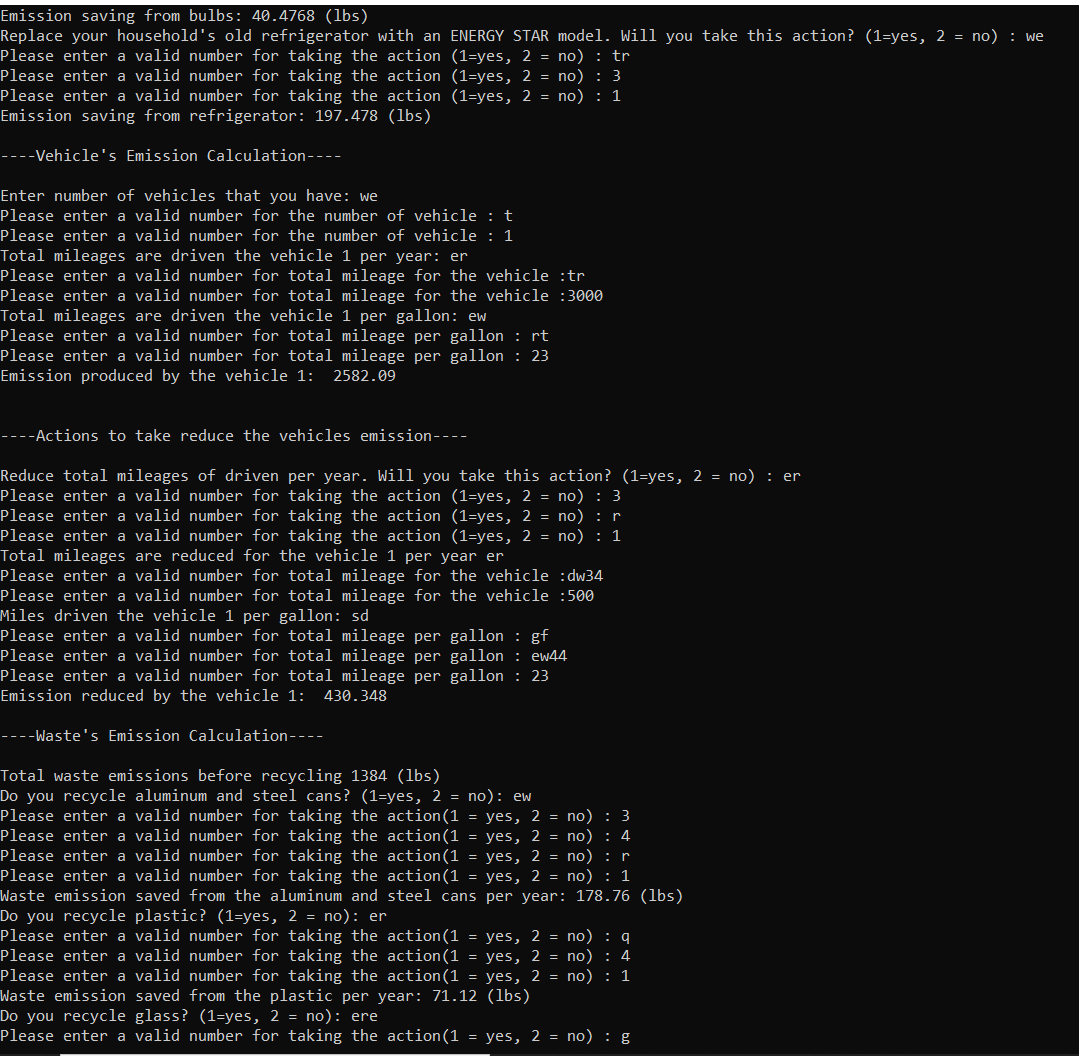
Screenshot # 5 is a snapshot of a CSV file from where we read the electricity emission factor for different zip codes. Screenshot #6 showed the sample output text file of the summary report.

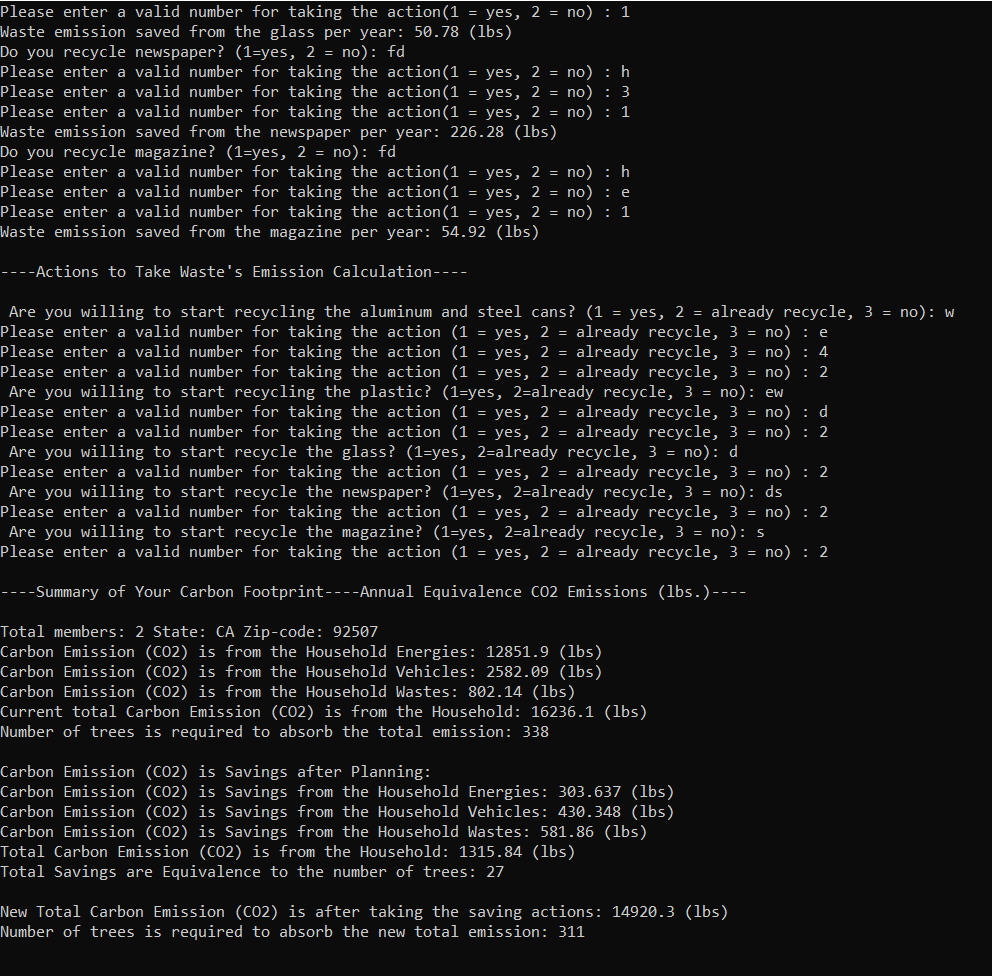
Screenshot # 1

Screenshot # 2

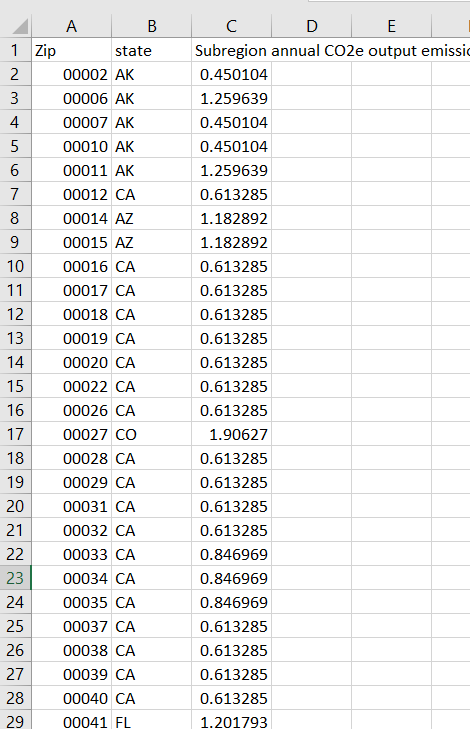
Screenshot # 3



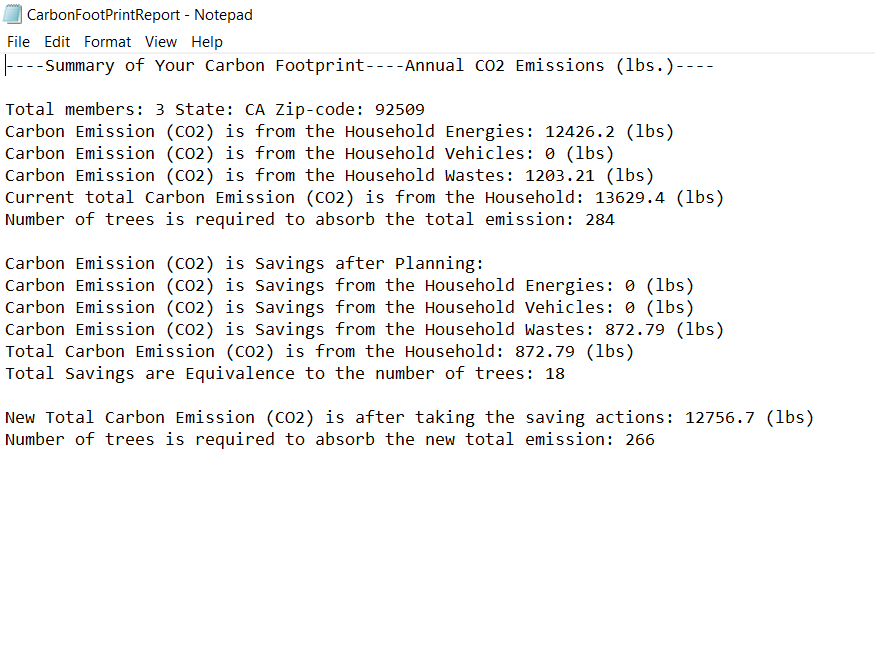




Screenshot # 4



Screenshot # 5



Screenshot # 6

**Programming Assignment Experience**

* This programming assignment was very interesting and learnable. We faced a little bit problem to select a project idea.
* We learnt many things from the project, for example, how to work in a group, program modularization, and how to reduce the total lines of coding, various equations to calculate the emission for different resources.
* In our reviews, we decided to refactor our program to several files, separating main program, utility functions, calculation functions, and constants. For better code readability, we also commented and separated our program into sections using comment separators.
* We were found the programming assignment just right, but deadline was bit of tight as it was a group project.
* We used the environmental protection agency (EPA) excel file to calculate the carbon footprint. We collected different standard constant values, emission factors, and several equations to calculate emission from the excel file.

* We enjoyed a lot throughout the whole project. The project is uploaded to [GitHub Repo](https://github.com/julschong/CarbonEmissionCalculator) for better revision tracking, sharing, and collaboration.
* All group members were active to complete the project. All members review the EPA excel file and cross-checked all constant values, equations, and emission factors. Then the project member decided the various user-defined functions to fulfill the project requirement and minimize coding. All members were participating in coding:
* main function, CSV file loading (ReadCSVFile()) and searching function (SearchZipCode) were developed by Khadiza Akter
* The user defined function for energy saving (EnergyEmissionSaving()) and write the summary report (WriteSummaryReport()) to text files were developed by the Alonso Ibarra
* The user defined functions to calculate various emission (CalculateVehicleEmission(), CalculatedEnergyEmission(), CalculatedWasteEmission()) were developed by the Meng Huan Lee

All project members involved to integrate the coding, testing, review the emission results. Project members finally wrote and reviewed the report as well.