Outcomes:

- Working with C# Mono as one of Imperative Programming Languages
- Understanding how C# Mono works and some of the difficulties
- Working with some of C# Mono features
- Working with files in C# Mono
- Solving problems using C# Mono
- Running and testing C# Mono codes

Scoring (465/565):

- (10 points) submission on Git with **10 or more commits**. Your Git history should represent the progress of your work.
- (72 points) Successful implementation of 3(undergraduate)/4(graduate) problems.
- (10 points) The execution time. Your program is efficient and well designed.
- (8 points) The style of your code including indentation, comments, etc.
- (3 bonus points) All bonus codes are done effectively.

Requirements:

- On your laptop, add a new folder inside your **CSE465_565** folder, and call it **Homework4**.
- Download **Hw4.cs**, **zipcodes.txt**, **zips.txt**, **cities.txt**, and **states.txt** from canvas and put them inside the **Homework4** folder.

Instructions:

1. In your project, you are free to create as many classes within the **Hw4.cs** file. However, ensure that the class **Hw4** is the only class that contains a **Main** method. This specific setup is essential in streamlining the grading process.
2. The programs will process the datafile **zipcodes.txt** which contains over 81K lines of text, one line for each geographic location in the US territories.
   - The first line of the file is the header line that explains each column of the data.
   - Except the first line, all other lines are tab-separated.
3. When your **Hw4.cs** is run, it should create the following output files:
     please ensure that the filenames are kept exactly as stated here
   - **CommonCityNames.txt**: contains all the common city names that appear in all of the states listed in **states.txt**.

- i. Each city in the file must be <u>unique</u>, no duplicate names allowed.
- ii. The cities should appear in **sorted** <u>order.</u>
- iii. The cities should appear <u>one per line</u>.
- ○ **LatLon.txt**: for each zip code listed in **zips.txt**, there will be a corresponding line of output. Each output line will list the zip code's latitude and longitude.
  - i. The latitude and longitude must be separated <u>by a space</u> on each line.
  - ii. If a zip code has multiple entries, provide <u>the first one</u> listed in zipcodes.txt.
- ○ **CityStates.txt**: For each city listed in **cities.txt**, there will be a corresponding line of output. Each output line will list the states containing that city name.
  - i. The states must be separated <u>by a space</u>.
  - ii. The states should appear in **sorted** <u>order</u>.
  - iii. Each state should be listed <u>only once</u> for each city.
- ○ **grad.txt**: **(CSE565 only)** Display each of the following pieces of information - one per line
  - i. The two geographically most distant zipcodes - space separated
  - ii. The zip code with the largest population
  - iii. The zip code with the smallest population
  - iv. The zip code with the largest per capita wages
  - v. The zip code with the smallest per capita wages
  - vi. The city with the largest total population (e.g., Brooklyn, NY).
4. Clarity and correctness is the primary consideration when writing your codes.
5. The efficiency and performance is also another concern; Using obviously wasteful and brute force approaches will result in high elapsed time and loss of points.
6. You can compile and execute your code via Command Line (CMD)/Terminal using the following commands:
   - ○ mcs   *.cs (compile)
   - ○ mono   Hw4.exe (execute)

   you must use C# Mono for this homework
7. Commit and push your work right after any small progress. Your Git should demonstrate the progression of your work.
   - ○ Adding or removing empty lines or spaces in one commit, does not add authenticity to your work.
   - ○ Committing and pushing large volumes of code within a very short time frame, such as minutes or seconds apart, can raise doubts about the work's originality. Producing significant amounts of code this quickly is unrealistic and may indicate that the code is copied from another source. Genuine coding usually requires more development time and deliberate, thoughtful submission of changes.
8. The execution time and style of your code, including indentation and comments, should be acceptable. Brute force approaches which lead to very slow programs, and a lack of clarity will result in a loss of points.

Bonus Points:

To earn 3 bonus points on your assignment, you need to implement the following features in C#:

1. Pointers/unsafe/fixed. Use pointers in your code where it makes sense and helps with the performance of your code.
2. Delegate function. Use Delegate functions in your code where it makes sense and helps with the performance of your code.
3. Lambda. Use Lambda functions in your code where it makes sense and helps with the performance of your code.

If you used any of the above features, you must specify where in your code you used them. At the very beginning of your **Hw4.cs** file, add special comments (the format is explained below) to point out exactly where in your code you've used pointers.

The following is an example of the comments you need to include at the top of your code

```
/*
1- => Used Pointers/unsafe/fixed in line/s: 10-15, 40-63
2- => Used Delegate in line/s: 17-20, 80
3- => Used Lambda in line/s: 65, 89, 104
*/
```

- Failing to highlight the use of bonus features by not adding the required comments at the top of your file will result in 0 bonus points.

Test your program:

Come up with more tests and test your program comprehensively. Your code will be tested separately with different text files.

Proper testing requires time. Begin by considering various scenarios and delving into the specifics. Eventually, you'll need to come up with a way to validate your solutions using the data at hand. Remember, thorough testing demands quality time devoted at the end to ensure your code functions as intended. So, as you manage your schedule for this assignment, make sure to allocate sufficient time for testing. <u>The consequence of lack of proper testing will result in losing points.</u>

Submission:

Submit the GitHub/GitLab URL of the project **Homework4**.
Inside this folder there should be the following files:

1. **Hw4.cs**.
2. All the input and output text files.