Problem 1: Extract even rows from CSV file and write to a new CSV.

Modify the CSV file **exampleWithHeader.csv** so that it **skips every even row** (rows 2, 4, 6, 8, etc.) and **writes the remaining rows** to a new CSV file, **output.csv.**

🎯 **Important suggestion:** Use the tools and techniques we've learned so far (like `csv.reader`, `csv.writer`, `append()` , `writerow()` etc).

Problem 2: Extract headers only from CSV files and write to a new folder

Using the `csv` and `os` modules, write a program that:

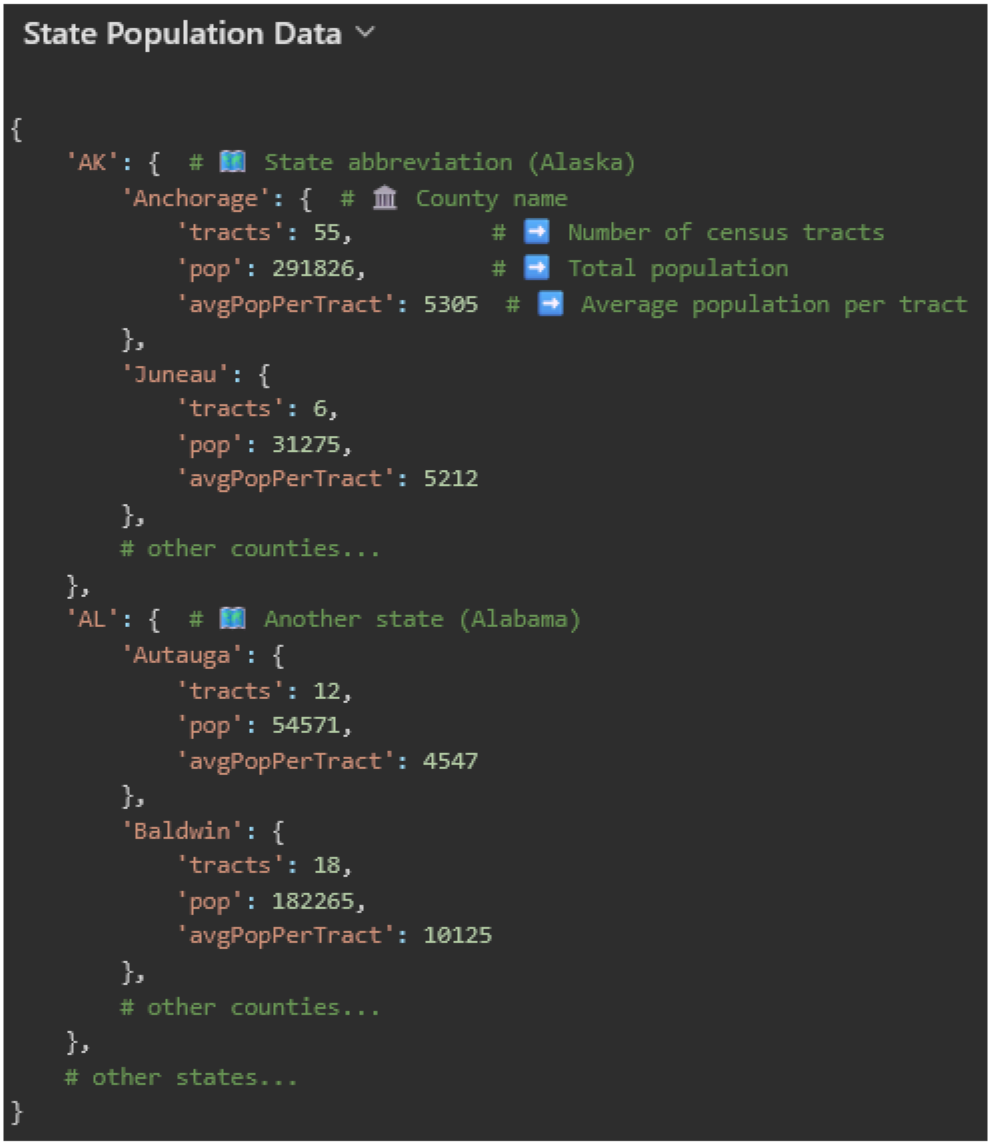
- Create a new folder called `headerOnly/`
- Loops through all `.csv` files in the zip file named removeCsvHeaders.zip
- Extract **only the header row** from each file
- Save each header into a new `.csv` file inside `headerOnly/`, using the **same filename** as the original.

🎯 **Important suggestion:** You are recommended to use `csv.reader()` and basic file handling (`open`, `with`, `newline='', line_num etc`) that we have covered in the class.

Problem 3: Add average population per tract to countyData Dictionary

Open the censusdata.xlsx file we covered in class and modify the countyData dictionary structure we used in our class code, so that besides population and tracts, you *ALSO STORE* the average population per tract for each county.

The dictionary should look like the image below.



State Population Data ⌄

```
{
    'AK': {  # 🗺 State abbreviation (Alaska)
        'Anchorage': {  # 🏛 County name
            'tracts': 55,          # ➡️ Number of census tracts
            'pop': 291826,         # ➡️ Total population
            'avgPopPerTract': 5305  # ➡️ Average population per tract
        },
        'Juneau': {
            'tracts': 6,
            'pop': 31275,
            'avgPopPerTract': 5212
        },
        # other counties...
    },
    'AL': {  # 🗺 Another state (Alabama)
        'Autauga': {
            'tracts': 12,
            'pop': 54571,
            'avgPopPerTract': 4547
        },
        'Baldwin': {
            'tracts': 18,
            'pop': 182265,
            'avgPopPerTract': 10125
        },
        # other counties...
    },
    # other states...
}
```

Save the updated dictionary to a .py file so that it can be imported and called, similar to the census2010.py file we created in clas.

🎯 **Important suggestion:** You may try adding `'avgPopPerTract'` **after** processing all rows (otherwise you'd divide before having the final totals!).