Coding Challenge - Abalone

In Grover we are working on streaming data processing. In this task we would like you to write a streaming application in a language of choice (preferably Rust or Python). The data file `abalone_full.csv` contains marine gastropod called Abalone's features.

Please solve the challenge below by meeting the given acceptance criteria. When you have finished please ZIP your solution and do not publish your solution in any public GitHub repository or anywhere else!

In general, feel free to make assumptions and have fun with the challenge! Please make sure you focus your time on explaining and elaborating your thought process, assumptions, decisions, thoughts and comments, for example inside a `thought-process.md` file. Please also document how to build and run your code.

Data features

```
Name / Data Type / Measurement Unit / Description
-----------------------------
Sex / nominal / -- / M, F, and I (infant)
Length / continuous / mm / Longest shell measurement
Diameter / continuous / mm / perpendicular to length
Height / continuous / mm / with meat in shell
Whole weight / continuous / grams / whole abalone
Shucked weight / continuous / grams / weight of meat
Viscera weight / continuous / grams / gut weight (after bleeding)
Shell weight / continuous / grams / after being dried
Rings / integer / -- / +1.5 gives the age in years
```

Acceptance Criteria

- We want to have streaming application which will produce CSV files according to these rules:
  - `infants_with_more_than_14_rings.csv` which will have all infants who have 14 or more rings. For all infants exactly the same number of rings, we want to enumerate and give an `id` to them. Final CSV will contain also `id` column.
    - To make the understanding clear; let's say, there are 5 infants with 13 rings we want to give an id to them like: 0,1,2,3,4. Then let's say we have 3 infants with 12 rings. Then we will have 0,1,2 as their id in the csv.
  - `males_heavy_and_short.csv` which will have all the males with `Whole_weight` more than `0.4` and `Length` is less than `0.5`. Please consider adjustable values for these constraints in your design. Like in the previous task since there is no count needed, you can enumerate all elements with incrementing `id` field.
  - **Optional:** `shell_humidity.csv` will contain `(Whole_weight - Shucked_weight) - Shell_weight` for every abalone to get the idea of how much water is absorbed in the abalones' shell. This feature will be added as a separate column called `shell_humidity_weight`. Data extracted to this column cannot be negative, and all rows which computes negative value should be omitted from the output csv. In addition to that, `id` field is useful to have in the output CSV file.

- Make three processing tasks above concurrent if possible.
- Output CSV files will contain all fields in the source data and `id` field.

Tech Stack

- Rust or Python (others also possible)
- No external databases, do all processing in memory to avoid complexity.
- Please don't use machine learning libraries like (e.g. scikit, pandas).
- Application should conform streaming mentality.
- Test code where youori feel it makes sense