

Fetch Coding Exercise - Software Engineering Internship

What do I need to submit?

Please write a program that reads from a CSV file called **transactions.csv**

(<https://fetch-hiring.s3.amazonaws.com/transactions.csv>), in the current working directory, processes the arguments, and

returns a response based on the conditions outlined in the next section. You can use any language.

We must be able to compile and run your code; provide any documentation necessary to accomplish this as part of the code you submit. Please assume the reviewer has not executed code in your language before when writing your README.

Assignment #1

Background

Our users have points in their accounts. Users only see a single balance in their account. But for reporting purposes, we actually track their points per payer. In our system, each transaction record contains: **payer** (string), **points** (integer), **timestamp** (date).

For earning points, it is easy to assign a payer. We know which actions earned the points. And thus, which partner should be paying for the points.

When a user spends points, they don't know or care which payer the points come from. But, our accounting team does care how the points are spent. There are two rules for determining what points to "spend" first:

- We want the oldest points to be spent first (oldest based on transaction timestamp, not the order they're received)
- We want no payer's points to go negative.

We expect your code to

1. Read the transactions from a CSV file.
2. Spend points based on the argument using the rules above.
3. Return all payer point balances.

Example

1. When you run your program, you will pass in 2 arguments, 1) which is the amount of points to spend 2) name of CSV file

For example, if you are using Python (you don't have to use Python) to spend 5000 points, it would look like this:

```
python3 mycode.py 5000 transactions.csv
```

2. Your code will ingest a CSV file with an example sequence.

- "payer", "points", "timestamp"
- "DANNON", 1000, "2020-11-02T14:00:00Z"
- "UNILEVER", 200, "2020-10-31T11:00:00Z"
- "DANNON", -200, "2020-10-31T15:00:00Z"
- "MILLER COORS", 10000, "2020-11-01T14:00:00Z"
- "DANNON", 300, "2020-10-31T10:00:00Z"

3. After the points are spent, the output should return the following results:

```
{
  "DANNON": 1000,
  "UNILEVER": 0,
  "MILLER COORS": 5300
}
```

Assignment #2

Please answer the following questions and paste your answers into a file called **summary.txt**. This file should be located in the same repository as your code.

1. Why did you choose the tools, libraries, and language you used for the coding exercise?
2. What are the advantages and disadvantages of your solution?
3. What has been a favorite school/personal project thus far? What about it that challenged you?

How do I submit it?

Provide a link to a public repository, such as GitHub or BitBucket, that contains your code via the provided link through Greenhouse. Any submissions in another format (.zip for example) will not be reviewed by our team.

FAQ

How will this exercise be evaluated?

An engineer will review the code you submit. At a minimum they must be able to run the code and the code must provide the expected results. You should provide any necessary documentation within the repository.

There should not be any commented out debug code (~~#Print("Inside the loop")~~) left in the final draft, everything should be formatted nicely and spelled correctly. Humans are grading this, not machines. Impress us!

I have questions about the problem statement

For any requirements not specified via an example, use your best judgment.

Can I provide a private repository?

If at all possible, we prefer a public repository because we do not know which engineer will be evaluating your submission. Providing a public repository ensures a speedy review of your submission. If you are still uncomfortable providing a public repository, you can work with your recruiter to provide access to the reviewing engineer.