

Software Assessment - Work Orders

This assessment will be evaluated based on the following criteria:

- Completion: Does your solution work for different inputs?
- Fundamentals: Is your code efficient? Do you use the appropriate data structures to store information?
- Code Quality: How organized is your solution and the code easy to read?
- Testing: Did you write some tests for the solution? How is the coverage?
- Speed: We are looking for solutions that are completed within an appropriate time frame.
- Communication: We are looking for candidates with strong communication skills. This will be evaluated based on your Readme.md file.

Goal

In this assessment, you will be working with *work orders*. Each order has an id, a name, and dependencies (orders can depend on the completion of other orders).

The goal of this assessment is to generate a file that shows all work orders and their dependencies.

Inputs

[Here is a compressed folder with an example of the two input files.](#)

orders.txt

This file contains information about all the orders. The file has the following columns:

- **id** - the id of the order (any integer between 0 and 10000)
- **name** - the name of the order (any string of length 1 to 100)

Here is the content of the example input:

```
id,name
1,Order 1
2,Order 2
3,Order 3
4,Order 4
5,Order 5
6,Order 6
```

dependencies.txt

This file contains all dependency relationships. Each row represents one dependency (i.e. order 2 is a dependency of order 1 in the example below). The file has the following columns:

- **id** - the id of an order
- **dependency_id** - the order id that is a dependency of order id in column 1

Here is the content of the example input:

```
id,dependency_id
1,2
1,3
3,4
5,2
3,6
```

Output

The expected output of the program is a text file that shows all the work orders and their dependencies.

Note:

- **At the root (first) level, only the orders that are not dependent on any other order are outputted**
- Underneath each order, show all the dependencies for that order (the dependencies do not need to be sorted)
- Orders can be shown multiple times

Below is an example output given the [example inputs](#) and here is a [text file version](#) of the example output.

```
Id: 1, Name: Order 1
Dependencies
  Id: 2, Name: Order 2
  Id: 3, Name: Order 3
  Dependencies
    Id: 4, Name: Order 4
    Id: 6, Name: Order 6
Id: 5, Name: Order 5
Dependencies
  Id: 2, Name: Order 2
```

And here is another set of [inputs and output](#). You may want to use it for testing your submission after you complete it.

Readme

Along with your submission, write a Readme.md file that:

- How to run the application:
 - The command we should use installing your dependencies, e.g: *npm install, pip install -r requirements.txt*. You can assume we already have your programming language set up in our environment, as well as any common package managers (npm, yarn, bundle, pip, etc)
 - The command we should use for actually running the application, e.g: *npm start, java Main, python manage.py serve*
- sentences) high-level description of your project, written for a person. This will help us assess how well you can communicate.

Submission Details

Please submit your code in a compressed folder (.zip, .sitx, .7z, .rar, and .gz) on the [Hatchways platform](#). The max submission size is 5MB.

Do not submit any built folders, since the compressed folder will be too large.

If your submission is too big (although this should not be the case if you follow the steps above), and you can't figure out how to compress, you are welcome to email your solution to hello@hatchways.io.

Please include your name, and use the email you signed up with the Hatchways platform. Use the subject line "Backend Assessment Submission".