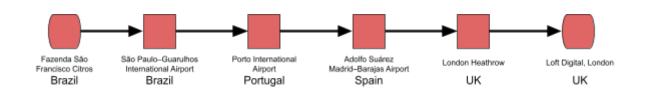




### **Task**

The Loft Digital coding monkeys have finished all their bananas, so they ordered a 400-kilogram crate of fair trade bananas which were transported from Brazil to London.

To save money, the banana crate was transported using several different delivery companies, each taking it a different step of the way from Brazil to London.



Your client wants to know the exact journey that the banana crate took, and says they will supply you with a set of notes describing every step of the journey; however, they can't guarantee what order the notes will be in. They could be completely random.

Therefore, your task is as follows:

Write a PHP application that can accept a set of journey steps in any order, and output the journey in the correct order.

## **Task details**

A **journey step** has the following JSON schema:

```
{
    "type": "object",
    "properties": {
        "from": {"type": "string"},
        "to": {"type": "string"}
    }
}
```

Here is an example journey step:

```
{
    "from": "Fazenda São Francisco Citros, Brazil",
    "to": "São Paulo-Guarulhos International Airport, Brazil"
}
```

Your application must be able to accept an JSON-formatted array of journey steps *in any order*, and output an JSON-formatted array of all locations visited *in the correct order*.

NOTE: You don't need to create a web application or a networked API; a command-line PHP application is fine.

## **Example**

#### Input

```
[
    "from": "Adolfo Suárez Madrid-Barajas Airport, Spain",
    "to": "London Heathrow, UK"
},
    "from": "Fazenda São Francisco Citros, Brazil",
    "to": "São Paulo-Guarulhos International Airport, Brazil"
},
    "from": "London Heathrow, UK",
    "to": "Loft Digital, London, UK"
},
    "from": "Porto International Airport, Portugal",
```

```
"to": "Adolfo Suárez Madrid-Barajas Airport, Spain"
},
{
    "from": "São Paulo-Guarulhos International Airport, Brazil",
    "to": "Porto International Airport, Portugal"
}
]
```

#### **Expected output**

```
[
    "Fazenda São Francisco Citros, Brazil",
    "São Paulo-Guarulhos International Airport, Brazil",
    "Porto International Airport, Portugal",
    "Adolfo Suárez Madrid-Barajas Airport, Spain",
    "London Heathrow, UK",
    "Loft Digital, London, UK"
]
```

# Requirements

- The implementation of your sorting algorithm must work with *any* set of journey steps (as defined by the supplied JSON schema). You can assume that there will always be an unbroken chain between all the steps of the trip.
- No third-party frameworks are allowed, apart from those required for testing. Start all code from scratch.
- You are allowed to use Composer for requiring dependencies necessary for testing, and for PSR-4 Autoloader functionality.
- Use Git to track your code changes.
- Provide a README.md file containing clear, simple instructions on how to execute the code and tests.
- The algorithm should have the lowest possible order of complexity you can manage ie. it should not take an unfeasibly long time/amount of resources to complete.
- If anything needs clarification which is not detailed in the instructions, don't panic; just make an assumption, and note it in the **README.md** file.

# What we are looking for

This task is designed to give us an idea of:

- Your ability to deliver an appropriate, simple solution to a given problem.
- How you structure your code.

- Your understanding of OOP.
- Your understanding of unit testing and/or TDD.
- Your ability to use version control.
- The efficiency of the sorting algorithm you implement.

# What we are not looking for

We are not looking for a solution designed to demonstrate your entire knowledge of PHP. Instead, make appropriate use of the language to solve the problem in hand cleanly.