

Factory Controller

Coding Exercise

The purpose of this exercise is to develop a software program meeting the requirements of the task. The objective is to review:

- Coding cleanliness and hygiene
- Approach to problem solving
- Use of design patterns and algorithms
- Approach to testing and code control

You are welcome to over-engineer parts of the solution in order to demonstrate an understanding of advanced patterns, or even alternative approaches to the problem.

The final product returned should include:

- A link to a git repository, or a tarball/zip file *containing a git repository*, which includes:
 - The working software program
 - Unit tests
 - A history of commits demonstrating your approach
- A copy of the program's output (text based) or screenshots of a running GUI application

The program may be written in any language; it may be a CLI application, a web application or a GUI application - however, where possible, try to make the application platform-agnostic and easy to be run on a reviewer's machine.

The Task

You have a factory. It is a great factory that can build a number of things. It can take components from an inventory and build a bigger component, returning the larger component to the inventory.

Your task is to build a system that will instruct the factory what sub-components to build first, in order to complete a larger order.

You will be provided a starting inventory list and a “recipe” schema list, a recipe schema contains:

- A list of items required to build this recipe
- A list of items that are built as a result (it's not always a single item)
- The time it takes to build this recipe

If your factory cannot build all required sub-components for a build order, it should not build any of the sub-components, but instead fail that build task.

If, for example, you are asked to build an *electronic circuit*, this circuit first requires *copper cables*. Your program will detect that you have no copper cables in the inventory and will first instruct the factory to build the cables *before* attempting to build the electronic circuit.

Once the program is complete, give your factory these build orders (in order):

- 3x `electric_engine`
- 5x `electric_circuit`
- 3x `electric_engine`

The program should first build all required parts to complete each item, however it will also run out of resources and will not be able to complete the final `electric_engine`.

- Some of the 5x electric circuits should be consumed by the final build order
- Only 4 of the 6 electric engines will be built - the 5th will fail due to a lack of iron plates (and consume no resources)

You should also display the time it took to build each build order. This may vary, depending on the number of sub-components it requires to be built.

Output the final state of the inventory once the program has completed.

Supporting content:

- [Appendix A](#): JSON file containing the starting inventory
- [Appendix B](#): JSON file containing recipe schemas
- [Appendix C](#): Sample program output, showing expecting outcomes

Appendix A

inventory.json

```
{  
  "iron_plate": 40,  
  "iron_gear": 5,  
  "copper_plate": 20,  
  "copper_cable": 10,  
  "lubricant": 100  
}
```

Appendix B

recipes.json

```
{
  "recipe_gear": {
    "title": "Gear",
    "time": 0.5,
    "consumes": {
      "iron_plate": 2
    },
    "produces": {
      "iron_gear": 1
    }
  },
  "recipe_pipe": {
    "title": "Pipe",
    "time": 0.5,
    "consumes": {
      "iron_plate": 1
    },
    "produces": {
      "pipe": 1
    }
  },
  "recipe_cables": {
    "title": "Copper Cable (2x)",
    "time": 0.5,
    "consumes": {
      "copper_plate": 1
    },
    "produces": {
      "copper_cable": 2
    }
  },
  "recipe_steel": {
    "title": "Steel Plate",
    "time": 16.0,
    "consumes": {
      "iron_plate": 5
    },
    "produces": {
      "steel_plate": 1
    }
  },
  "recipe_circuit": {
    "title": "Electric Circuit",
    "time": 1.5,
    "consumes": {
      "iron_plate": 1,
      "copper_cable": 3
    },
    "produces": {
      "electric_circuit": 1
    }
  }
}
```

```
},
"recipe_engine_block": {
  "title": "Engine Block",
  "time": 10.0,
  "consumes": {
    "steel_plate": 1,
    "iron_gear": 1,
    "pipe": 2
  },
  "produces": {
    "engine_block": 1
  }
},
"recipe_elec_engine": {
  "title": "Electric Engine",
  "time": 10.0,
  "consumes": {
    "electric_circuit": 2,
    "engine_block": 1,
    "lubricant": 15
  },
  "produces": {
    "electric_engine": 1
  }
}
}
```

Appendix C

Sample Program Output

Inventory loaded: 5 unique components

Recipes loaded: 7 total

INVENTORY:

- * iron_plate: 40
- * iron_gear: 5
- * copper_plate: 20
- * copper_cable: 10
- * lubricant: 100

```
> building recipe 'recipe_circuit' in 1.5s (1.5s total)
> building recipe 'recipe_circuit' in 1.5s (1.5s total)
  > building recipe 'recipe_steel' in 16.0s (16.0s total)
    > building recipe 'recipe_pipe' in 0.5s (0.5s total)
      > building recipe 'recipe_pipe' in 0.5s (0.5s total)
    > building recipe 'recipe_engine_block' in 10.0s (27.0s total)
  > building recipe 'recipe_elec_engine' in 10.0s (40.0s total)
Built electric_engine in 40.0 seconds
```

```
> building recipe 'recipe_circuit' in 1.5s (1.5s total)
  > building recipe 'recipe_cables' in 0.5s (0.5s total)
> building recipe 'recipe_circuit' in 1.5s (2.0s total)
  > building recipe 'recipe_steel' in 16.0s (16.0s total)
    > building recipe 'recipe_pipe' in 0.5s (0.5s total)
      > building recipe 'recipe_pipe' in 0.5s (0.5s total)
    > building recipe 'recipe_engine_block' in 10.0s (27.0s total)
  > building recipe 'recipe_elec_engine' in 10.0s (40.5s total)
Built electric_engine in 40.5 seconds
```

```
  > building recipe 'recipe_cables' in 0.5s (0.5s total)
    > building recipe 'recipe_cables' in 0.5s (0.5s total)
  > building recipe 'recipe_circuit' in 1.5s (2.5s total)
    > building recipe 'recipe_cables' in 0.5s (0.5s total)
  > building recipe 'recipe_circuit' in 1.5s (2.0s total)
    > building recipe 'recipe_steel' in 16.0s (16.0s total)
      > building recipe 'recipe_pipe' in 0.5s (0.5s total)
        > building recipe 'recipe_pipe' in 0.5s (0.5s total)
      > building recipe 'recipe_engine_block' in 10.0s (27.0s total)
    > building recipe 'recipe_elec_engine' in 10.0s (41.5s total)
Built electric_engine in 41.5 seconds
```

```
> building recipe 'recipe_cables' in 0.5s (0.5s total)
  > building recipe 'recipe_cables' in 0.5s (0.5s total)
> building recipe 'recipe_circuit' in 1.5s (2.5s total)
Built electric_circuit in 2.5 seconds
```

```
> building recipe 'recipe_cables' in 0.5s (0.5s total)
> building recipe 'recipe_circuit' in 1.5s (2.0s total)
Built electric_circuit in 2.0 seconds
```

```
> building recipe 'recipe_cables' in 0.5s (0.5s total)
```

```
> building recipe 'recipe_cables' in 0.5s (0.5s total)
> building recipe 'recipe_circuit' in 1.5s (2.5s total)
Built electric_circuit in 2.5 seconds
```

```
> building recipe 'recipe_cables' in 0.5s (0.5s total)
> building recipe 'recipe_circuit' in 1.5s (2.0s total)
Built electric_circuit in 2.0 seconds
```

```
> building recipe 'recipe_cables' in 0.5s (0.5s total)
> building recipe 'recipe_cables' in 0.5s (0.5s total)
> building recipe 'recipe_circuit' in 1.5s (2.5s total)
Built electric_circuit in 2.5 seconds
```

```
> building recipe 'recipe_steel' in 16.0s (16.0s total)
> building recipe 'recipe_pipe' in 0.5s (0.5s total)
> building recipe 'recipe_pipe' in 0.5s (0.5s total)
> building recipe 'recipe_engine_block' in 10.0s (27.0s total)
> building recipe 'recipe_elec_engine' in 10.0s (37.0s total)
Built electric_engine in 37.0 seconds
```

Insufficient resources to build: electric_engine

INVENTORY:

```
* iron_plate: 1
* iron_gear: 1
* copper_plate: 8
* copper_cable: 1
* lubricant: 40
* electric_circuit: 3
* steel_plate: 0
* pipe: 0
* engine_block: 0
* electric_engine: 4
```