Technical challenge - Software Engineer

Challenge description

Bloomon has a production facility that produces bouquets. We simplified how the real one (located in Amstelveen) works, for the purpose of this technical challenge:

- It uses *flowers* of different *species* and *sizes* as *input*;
- It produces *bouquets* according to *design* specifications as *output*;
- The *flowers* arrive into the facility one-by-one, and they can be stored there until there are enough *flowers* to create a *bouquet*.

Your job is to create a command line application (in Python) that takes the *design* specifications and the stream of *flowers* as an *input*, and produce the stream of *bouquets* as an *output*.

The solution should have all configuration files needed to be built and run in a Docker container (don't expect anything else but docker to be installed).

Completing the challenge should take approximately 4 hours and we expect you to return it within a week.

We are going evaluate both the correctness of your solution as well as it's design and code quality.

Good luck!

Input / output specifications

- The solution needs to work with standard input and output (stdin & stdout).
- The ***input*** contains ***designs*** to be produced and available ***flowers***:

```
design1
design2
...
<empty line>
flower1
flower2
flower3
...
```

- The ***output*** should be a ***bouquet*** every time one can be created from the available ***flowers***:

```
bouquet1
bouquet2
...
```

Data specifications

- A ***flower species*** is identified by a single, lowercase letter: `a - z`.
- A ***flower size*** is indicated by a single, uppercase letter: `L` (large) and `S` (small).
- A ***flower*** is identified by a ***flower species*** and a ***flower size***: for example, `rL`.
- A ***design name*** is indicated by a single, uppercase letter: `A - Z`.
- A ***design*** is single line of characters with the following format:

```
<design name><flower size><flower1 max quantity><flower1
species>...<flowerN max quantity><flowerN species><total quantity>
```

- The format includes ***flower* size** only once and it defines the size for all flowers in the given design (i.e. a large *design* can only have large *flowers).*
- The ***flower species*** are listed in alphabetic order and only appear once***.***
- The ***flower max quantities*** are always larger than 0***.*** The ***flower min quantities*** are implicit and always equal to 1 (for all specified species).
- The **total quantity** of flowers can be smaller than the sum of the **flower max quantities** - allowing for some variation between required flower species.
- Example***:*** **`AL1d2r3t5`**

- A ***bouquet*** is single line of characters with the following format:

```
<design name><flower size><flower1 quantity><flower1 species>...
<flowerN quantity><flowerN species>
```

- The format includes ***flower* size** only once and it defines the size of all flowers in the given bouquet (i.e. a large *bouquet* can only have large *flowers).*
- The ***flower species*** are listed in alphabetic order and only appear once***.***
- The ***flower quantities*** are always larger than 0***.***
- Example***:*** **`AL1d2r2t`**

- A **`bouquet`** `must comply to its` **`design:`**
  - A ***bouquet*** must have all and only ***flower species*** required by the corresponding ***design*** (i.e. comply with the implicit ***flower min quantities***)***.***
  - Every required **flower *species*** in a ***bouquet*** must be in the ***flower quantity*** that is less or equal to the ***flower max quantity*** spe cified by the ***design***.
  - The sum of the ***flower quantities*** in a ***bouquet*** should be equal to the ***total quantity*** of flowers in the corresponding ***design.***

Example

The following ***input***

```
AS2a2b3
BL2a2

aL
bS
aS
bS
aS
aL
aS
bS
```

should produce the following ***output***

```
AS1a2b
BL2a
AS2a1b
```

Questions?

In case things aren't clear enough and/or not explicitly specified - please use your best judgment (but keep it simple). And don't forget to mention those in the readme!

Wrap up

Are you done? Great!! Please submit your solution in a private GitHub / GitLab repository and grant access to "BloomonDev" user.

Thank you for participating in our code challenge!