Developer Coding Challenge

Problem Description

Problem:

You are currently in your house wearing your PJ's. You must get fully dressed and leave the house.

Your challenge is to programmatically process a list of getting dressed commands, enforce related rules, and display appropriate output.

Input Format:

1. Temperature type (one of the following)
    - HOT
    - COLD
2. Comma separated list of numeric commands

Command	Description	Hot Response	Cold Response
1	Put on footwear	"sandals"	"boots"
2	Put on headwear	"sunglasses"	"hat"
3	Put on socks	fail	"socks"
4	Put on shirt	"shirt"	"shirt"
5	Put on jacket	fail	"jacket"
6	Put on pants	"shorts"	"pants"
7	Leave house	"leaving house"	"leaving house"
8	Take off pajamas	"Removing PJs"	"Removing PJs"

Regardless of how the program is invoked, the program must take the list of input commands as command line arguments to the program. For example:

```
./get-dressed HOT 1, 2, 3, 4, 5, 6, 7, 8
```

Rules:

- You start in the house with your PJ's on
- Pajamas must be taken off before anything else can be put on
- Only 1 piece of each type of clothing may be put on
- You cannot put on socks when it is hot
- You cannot put on a jacket when it is hot
- Socks must be put on before footwear
- Pants must be put on before footwear
- The shirt must be put on before the headwear or jacket
- You cannot leave the house until all items of clothing are on (except socks and a jacket when it's hot)
- If an invalid command is issued, respond with "fail" and stop processing commands

Examples

Success

Input: HOT 8, 6, 4, 2, 1, 7
Output: Removing PJs, shorts, shirt, sunglasses, sandals, leaving house

Input: COLD 8, 6, 3, 4, 2, 5, 1, 7
Output: Removing PJs, pants, socks, shirt, hat, jacket, boots, leaving house

Failure

Input: HOT 8, 6, 6
Output: Removing PJs, shorts, fail

Input: HOT 8, 6, 3
Output: Removing PJs, shorts, fail

Input: COLD 8, 6, 3, 4, 2, 5, 7
Output: Removing PJs, pants, socks, shirt, hat, jacket, fail

Input: COLD 6
Output: fail

Directions

Please submit your solution in Scala and provide all source, test, documentation, and build support files. The project structure is up to you, but assume that this code will be deployed to production and your peers will be maintaining the code going forward.

If you do not have professional Scala experience, we will take that into consideration during our assessment.

Please ZIP your files or we will not be able to receive them. You may instead publish your code to a private Github or Gitlab repository, or a file hosting service such as DropBox, Google Drive, etc.

Criteria

You will primarily be judged on the code directly related to the implementation of the stated problem and business rules:

- Correct implementation of business rules
- Code Legibility
- Testability
- Ease of Maintenance
- Use of functional programming concepts
- Use of recognizable best practices and patterns
- Submission of a working solution with basic usage instructions

Secondary evaluation criteria include the usage and evident knowledge of the tools, utilities, frameworks, and methodologies specified in the job description.

We value creativity and initiative to learn new technology; however, be advised that candidates that focus solely on the primary criteria will be more successful than candidates that focus instead on intricate interface or usage of a breadth of technologies.