Assignment: Package Challenge

Introduction

You want to send your friend a package with different things.

Each thing you put inside the package has such parameters as index number, weight and cost. The package has a weight limit. Your goal is to determine which things to put into the package so that the total weight is less than or equal to the package limit and the total cost is as large as possible.

You would prefer to send a package which weighs less in case there is more than one package with the same price.

Input sample

Your API should accept as its only argument a path to a filename. The input file contains several lines. Each line is one test case.

Each line contains the weight that the package can take (before the colon) and the list of items you need to choose. Each item is enclosed in parentheses where the 1st number is a item's index number, the 2nd is its weight and the 3rd is its cost. E.g.

```
81 : (1,53.38,€45) (2,88.62,€98) (3,78.48,€3) (4,72.30,€76) (5,30.18,€9)
(6,46.34,€48)
8 : (1,15.3,€34)
75 : (1,85.31,€29) (2,14.55,€74) (3,3.98,€16) (4,26.24,€55) (5,63.69,€52)
(6,76.25,€75) (7,60.02,€74) (8,93.18,€35) (9,89.95,€78)
56 : (1,90.72,€13) (2,33.80,€40) (3,43.15,€10) (4,37.97,€16) (5,46.81,€36)
(6,48.77,€79) (7,81.80,€45) (8,19.36,€79) (9,6.76,€64)
```

Output sample

For each set of items that you put into a package provide a new row in the output string (items' index numbers are separated by comma). E.g.

```
4
-
2,7
8,9
```

Constraints

You should write a class com.mobiquity.packer.Packer with a static API method named pack. This method accepts the absolute path to a test file as a String. The test file will be in UTF-8 format. The pack method returns the solution as a String.

Your method should throw an error/exception named **APIException** (com.mobiquity.packer.APIException where relevant) if any constraints are not met. Therefore your API signature in pseudocode should look like:

```
string pack(string filePath)
```

Additional constraints:

1. Max weight that a package can take is ≤ 100
2. There might be up to 15 items you need to choose from
3. Max weight and cost of an item is ≤ 100

Remember

Apply best practices for software design & development and document your approach (what strategy/algorithm/data structure/design pattern you chose and why) and put comments into your source files. We do consider TDD a best practice.

Your Result

When finished, please send a link to a public repository or a zip file to your contact person within Mobiquity. The zip file/repository should include the source files for your solution. The source code will be examined by one of our developers. Note that your delivered source code should be considered production release ready.

Your solution is meant to be used as a cross platform **library**, NOT as a standalone application.

Good luck with this assignment. If you have any questions, don't hesitate to ask your contact person within Mobiquity.