

Q1. Fizz Buzz

The original fizz-buzz consists in writing all numbers from 1 to 100, and just replacing all multiples of 3 by “fizz”, all multiples of 5 by “buzz”, and all multiples of 15 by “fizzbuzz”.

Your goal is to implement a web server that will expose a REST API endpoint that:

- Accepts five parameters : three integers int1, int2 and limit, and two strings str1 and str2.
- Returns a list of strings with numbers from 1 to limit, where: all multiples of int1 are replaced by str1, all multiples of int2 are replaced by str2, all multiples of int1 and int2 are replaced by str1str2.
- The output must look like this : ["1", "2", "fizz", "4", "buzz", "fizz", "7", "8", "fizz", "buzz", ...]

Add a statistics endpoint allowing users to know what the most frequent request has been. This endpoint should:

- Accept no parameter
- Return the parameters corresponding to the most used request, as well as the number of hits for this request
- The output must look like this :
{"int1":3,"int2":5,"limit":150,"str1":"fizz","str2":"buzz","count":3}

The server needs to be:

- Ready for production
- Easy to maintain by other developers
- Code must be in kotlin (or java) with Spring Boot

Precision :

L'algo est volontairement simple car l'accent sera mis sur la réalisation d'une api PROD ready. Du coup, libre à vous d'ajouter tout ce qu'il vous semble nécessaire afin que l'api soit utilisable en l'état.

Q2. XspeedIt

XspeedIt est une société d'import / export ayant robotisé toute sa chaîne d'emballage de colis. Elle souhaite trouver un algorithme permettant à ses robots d'optimiser le nombre de cartons d'emballage utilisés. Les articles à emballer sont de taille variable, représentés par un entier compris entre 1 et 9.

Chaque carton a une capacité de contenance de 10.

Ainsi, un carton peut par exemple contenir un article de taille 3, un article de taille 1, et un article de taille 6.

La chaîne d'articles à emballer est représentée par une suite de chiffres, chacun représentant un article par sa taille.

Après traitement par le robot d'emballage, la chaîne est séparée par des "/" pour représenter les articles contenus dans un carton.

Exemple :

Chaîne d'articles en entrée : 163841689525773

Chaîne d'articles emballés : 163/8/41/6/8/9/52/5/7/73

L'algorithme actuel du robot d'emballage est très basique.

Il prend les articles les uns après les autres, et les mets dans un carton.

Si la taille totale dépasse la contenance du carton, le robot met l'article dans le carton suivant.

Objectif :

Implémenter un algorithme qui permettrait de maximiser le nombre d'articles par carton, en utilisant Java ou Kotlin.

L'accent sera mis sur l'algorithme et le résultat. Il n'y a pas besoin de faire une api rest pour cet exercice. L'algo peut se trouver dans une classe de test ou dans une classe avec un main.

Précision :

Pour faciliter le dev, les articles sont sous forme d'un string "163841689525773" mais il faut garder en tête, quand on traite l'article n qu'on ne connaît pas les articles n+x (du coup ne pas trier la liste d'articles par exemple...). On ne connaît que l'article actuel et les articles précédents.

Exemple :

Articles : 163841689525773

Robot actuel : 163/8/41/6/8/9/52/5/7/73 => 10 cartons utilisés

Robot optimisé: 163/81/46/82/9/55/73/7 => 8 cartons utilisés