



Examen algorithmique G1 avec Java, 31 Mai 2022, 3h, internet non autorisé.

A. Problème algorithmique 1 : Loup-garou

Rakoto, est le chef de son village. Il sait que parmi ses villageois, se cache un loup garou (que l'on va abrégé en *loup* pour simplifier), qui tue ces derniers lorsque la nuit tombe. Il décide donc de monter la garde pour avertir les villageois de la proximité du loup, ou bien le chasser s'il le croise. Aidez Rakoto à prévenir le villageois devant le loup qu'il est sur le point d'être tué. Rappelez-vous que vous vous trouvez à l'avant de la file, qui se trouve à la fin du tableau :

[villageois, villageois, villageois, villageois, villageois, loup, villageois, villageois] (VOUS ÊTES ICI À L'AVANT)

7 6 5 4 3 2 1

Si le loup est le plus proche de vous, retournez "Va-t-en !". Sinon, retournez "Attention villageois numéro N !" où N est la position du villageois dans la file (de villageois).

Signature de la méthode :

```
public static String protectVillagers(String[] input)
```

Exemples :

Entrée : ["villageois", "villageois", "villageois", "villageois", "villageois", "loup", "villageois", "villageois"]

Sortie : "Attention villageois numéro 2! "

Entrée : ["villageois", "loup", "villageois", "villageois", "villageois", "villageois", "villageois"]

Sortie : "Attention villageois numéro 5! ";

Entrée : ["loup", "villageois", "villageois", "villageois", "villageois", "villageois", "villageois"];

Sortie: "Attention villageois numéro 6! "

Entrée : ["villageois", "loup", "villageois"];

Sortie : "Attention villageois numéro 1! "

Entrée : ["villageois", "villageois", "loup"];

Sortie : "Va-t-en !"

NB : Il y aura exactement un loup dans le tableau. Le loup ne figure pas dans le comptage du N-ème villageois.

B. Problème algorithmique 2 : Retirez les doublons

Dans ce Kata, vous recevrez un tableau de chaînes de caractères et votre tâche consiste à supprimer toutes les lettres en double consécutives de chaque chaîne dans le tableau.

Signature de la méthode :

```
public static String[] removeDup(String[] inputs)
```

Par exemple:

```
removeDup(["piccaninny", "hubbububboo"]) => ['picaniny', 'hubububo'];
removeDup(["abracadabra", "allottee", "assessee"]) => ['abracadabra', 'alote', 'asese'];
removeDup(["kelless", "keenness"]) => ['keles', 'kenes'];
removeDup(["adanac", "soonness", "toolless", "ppellee"]) => ['adanac', 'sones', 'toles', 'pele'];
removeDup(["callalloo", "feelless", "heelless"]) => ['calalo', 'feles', 'heles'];
removeDup(["putteellinen", "keenness"]) => ['putelinen', 'kenes'];
removeDup(["kelless", "voorraaddoospullen", "achcha"]) => ['keles', 'voradospulen', 'achcha'];
```

Les chaînes seront en minuscules uniquement, pas d'espace.

C. Problème algorithmique 3 : un semblant de déjà vu... les tests en moins

Merry et Pippin décident de participer à des compétitions lors d'un grand festival. Il y a une grande variété de compétitions, comme le tir à l'arc, le combat à l'épée, la chasse, l'équitation, etc. Cependant, ils réalisent qu'après avoir joué plusieurs tournois consécutifs, ils ont besoin de se reposer un jour. Ils recueillent des informations sur les prix en argent de chaque compétition. Une compétition a lieu chaque jour. Merry et Pippin décident qu'ils vont essayer de gagner le maximum de prix.

Le problème :

Compte tenu du prix de chaque tournoi, et du nombre maximum de jours consécutifs pendant lesquels Merry et Pippin peuvent jouer sans interruption, calculez le prix maximum qu'ils peuvent gagner.

Il y a N tournois en tout, organisés du jour 1 au jour N , représenté sous la forme d'un tableau. Merry et Pippin peuvent jouer au maximum un certain nombre de tournois consécutifs avant de devoir se reposer, que l'on va appeler *maxEndurance*. Vous devez retourner le montant total maximum que Pippin et Merry peuvent obtenir.

Signature de la méthode :

```
public static int getMaxPrizeSum(int[] prizes, int maxEndurance)
```

Exemples :

Soit les prix des tournois qui sont représentés par le tableau suivant, avec une *maxEndurance* de 4.

Tournoi	1	2	3	4	5	6	7	8	9	10
Prix	13	2	15	17	19	33	2	2	2	2

Merry et Pippin joueront donc les jours **1 > 3 > 4 > 5 > 6 > 8 > 9 > 10**. Ainsi, le prix maximum sera de 103 dans cet exemple ci.