

Battle dev HEI : Examen Final PROG1

Niveau 1 : Cérémonie d'ouverture

Cette année, vous avez l'honneur d'organiser les Jeux Olympiques. Et qui dit Jeux Olympiques dit cérémonie d'ouverture !

Vous avez opté pour une cérémonie très épurée : chaque délégation se déplace en ligne droite jusqu'à arriver devant le Comité Olympique.

On vous donne la distance initiale (en mètres) de chaque délégation au Comité Olympique ainsi que la vitesse (en mètres par seconde) de chaque délégation : pouvez-vous nous indiquer quelle délégation aura l'honneur de clôturer la cérémonie ? À noter que les délégations ne se gênent pas entre elles et peuvent se doubler sans perdre de temps.

Signature de la fonction : `getLastDelegation(delegations)`

- *delegations* est un tableau de chaîne de caractères. Chaque élément contient une chaîne de caractères, puis deux entiers positifs, séparés par des espaces. La chaîne de caractères indique le nom de la délégation, le premier entier indique sa distance au Comité Olympique (en mètres), et le deuxième la vitesse (en mètres par seconde) de la délégation.
- Sortie : Le nom de la délégation qui clôturera la cérémonie. On vous garantit qu'il n'y a qu'une seule délégation qui arrive en dernier (pas d'égalité possible).

Exemple. Pour l'entrée ["Chili 101 2", "Estonia 58 1", "Romania 102 3", "Ecuador 103 4", "Madagascar 104 5", "Oman 50 1", "Nicaragua 51 3", "Yemen 53 4"]

La sortie attendue est : "Estonia"

En effet, le Yemen arrive devant le Comité Olympique 13.25 secondes après le départ, suivi du Nicaragua 17 secondes après le départ. Peu après arrivent Madagascar (20.8 secondes après le départ) puis l'Équateur (25.75 secondes après le départ) et enfin la Roumanie (34 secondes après le départ). Bien après, Oman et le Chili atteignent le Comité Olympique au coude à coude (respectivement 50 et 50.5 secondes après le départ), et c'est l'Estonie qui clôt la cérémonie après 58 secondes !

Niveau 2 : Bobsleigh sur gazon

La célèbre épreuve de bobsleigh sur gazon approche à grands pas et vous devez aider les organisateurs à placer autant de bobsleigh que possible dans Central Park. Mais il y a un problème : le parc contient des arbres qui risquent de gêner la disposition des bobsleigh.

Vous disposez d'une carte (carrée) du parc sur laquelle les arbres sont représentés par des X et les espaces libres par des . et devez trouver le nombre maximum de bobsleighs que vous pouvez placer dans le parc. Les bobsleighs sont des segments horizontaux de 1x4 cases et ne peuvent pas être placés sur les arbres.

Quel est le nombre maximal de bobsleighs que vous pouvez placer dans le parc ? Notez bien que chaque bobsleigh doit être placé horizontalement : les règles du bobsleigh sur gazon sont impénétrables !

Signature de la fonction. countMaxBobsleigh(N, map)

- N est un entier représentant le nombre de lignes et de colonnes du terrain.
- map est un tableau composé de string à N caractères, avec des . sur les cases libres, et des X sur les cases contenant un arbre.
- Sortie attendue : le nombre maximum de bobsleigh que vous pouvez placer dans le parc.

Exemple. Pour l'entrée N = 5

```
map = [  
    "...X"  
    "X...",  
    "....",  
    "..X..",  
    "X...X"  
]
```

La sortie attendue est 3. En effet, on peut placer trois bobsleighs de la façon suivante (notez bien qu'on ne vous demande pas de renvoyer un tel placement, mais juste le nombre maximum de bobsleighs que l'on peut placer)

1111X

X2222

3333.

..X..

X...X

Niveau 3 : Traversée dans le Far West

Après avoir organisé la cérémonie d'ouverture et délimité la zone pour le bobsleigh sur gazon, vous décidez de prendre un peu de repos bien mérité. Mais c'est sans compter sur le Comité Olympique, qui vous confie une nouvelle mission : traverser le désert montagneux de l'Ouest sauvage pour aller chercher la torche olympique restée au poste frontière.

Heureusement, un vieux trappeur vous donne une carte avec des instructions de direction : «NORD», «SUD», «EST», «OUEST». Malheureusement, ces indications sont souvent contradictoires ou inutiles : aller au nord puis immédiatement au sud vous fait perdre un temps précieux... et dans ces conditions extrêmes, chaque pas de trop pourrait vous coûter la vie.

Votre objectif est donc de réduire le plan au strict minimum : chaque paire de directions immédiatement opposées doit être supprimée. Le but est de revenir au même point avec le moins de mouvements possible.

Signature de la fonction : `reduceDirections(directions)`

- `directions` est un tableau de chaînes de caractères contenant les mots "NORD", "SUD", "EST", "OUEST" (en majuscules).
- Retournez un tableau de chaînes de caractères représentant le plan simplifié : toutes les directions inutiles supprimées, dans le bon ordre.

Deux directions s'annulent si elles sont immédiatement opposées, même si cette simplification permet ensuite d'annuler d'autres paires plus éloignées.

Exemple 1. `reduceDirections(["NORD", "SUD", "SUD", "EST", "OUEST", "NORD", "OUEST"])`

Retour attendu : `["OUEST"]`

Exemple 2. `reduceDirections(["NORD", "SUD", "EST", "OUEST"])`

Retour attendu : `[]`