Test Java

Présentation :

Dans cet exercice, nous nous intéressons à un jeu dans lequel un joueur avance de case en case sur un plateau dans le but d'atteindre la case de fin.

Chaque case correspond à une action particulière. A chaque tour, le joueur effectue une action en fonction de la case sur laquelle il est tombé (jeter le dé pour avancer, avancer, ou reculer).

L'objectif du programme est de déterminer le nombre minimal de mouvements nécessaire au joueur le plus chanceux pour finir le jeu.

Plateau :

Nous allons considérer une version du jeu dans laquelle le plateau est linéaire, où les cases se suivent.

Chaque case peut contenir :

- Soit le caractère S (pour Start) qui représente la case de départ.
- Soit le caractère E (pour End) qui représente la case d'arrivée à atteindre.
- Soit le caractère R qui indique au joueur qu'il doit lancer le dé pour savoir de combien de cases avancer.
- Soit un nombre qui indique au joueur de combien de cases il doit avancer (si le nombre est positif) ou reculer (si le nombre est négatif).

Jeu :

Au début de la partie, le joueur commence sur la case de départ S.

Il commence en lançant le dé et avance du nombre de cases indiquées sur le dé. Les valeurs possibles du dé vont de 1 à 6.

Ensuite, il exécute l'action indiquée sur la case (lancer le dé, avancer, reculer).

Chaque action coute 1 tour.

Le jeu est gagné quand le joueur atteint la case d'arrivée E. Le joueur doit tomber exactement sur la case d'arrivée et ne peut pas la dépasser.

Les cases de départ et d'arrivée S et E ne se situent pas forcément au début et à la fin du plateau.

Un tour faisant sortir du plateau fait perdre le jeu.

Exemple :

Considérons le plateau suivant :

4	**S**	-2	1	R	4	3	4	3	-5	2	-4	**E**

Le joueur commence sur la case S. Le moyen le plus rapide de gagner est de :

- Lancer le dé et faire un 3, pour arriver sur la case R (tour 1).
- Relancer le dé et faire un 6, pour arriver sur la case 2 (tour 2).
- Le joueur est obliger d'avancer de 2 cases et arrive sur la case E (tour 3).

Le résultat attendu est donc **3**, car il faut 3 tours au minimum pour finir le jeu.

Objectif :

Vous devez écrire un programme qui trouve le nombre minimum de tours nécessaire pour gagner le jeu à partir d'un plateau donné. Si le plateau ne permet pas de finir le jeu, le résultat doit être -1.

Vous devez implémenter la fonction `compute` située dans la classe `solution.Solution`.

Cette fonction prend en paramètre un `BufferedReader` et renvoie un entier qui représente le résultat attendu. Le `BufferedReader` contient une série de lignes, dont la première est le nombre de cases du plateau, et les suivantes les intitulés des cases.

Exemple :

Input	Output
11	3
S	
1	
R	
4	
3	
4	
3	
-5	
2	
-4	
E	

Vous pouvez tester votre programme en lançant le `main` situé dans la classe `Solution`. Les tests sont déjà implémentés et vous pouvez changer le test à effectuer en changeant la valeur de la variable `testCaseID`, qui peut aller de 1 à 15, ou en lui assignant la valeur "ALL" pour exécuter tous les tests.

Seul le contenu de la fonction `compute` doit être développé.

Bien que pris en compte, nous donnerons moins d'importance au temps pris pour le développement, qu'à la qualité du code et à ses performances.