

Le bison zombie

1 le background

Le cinéma abonde en films et série sur les zombies qui ont presque tous le même pitch peu original, à savoir un ou plusieurs humains en situation de survie, devant régulièrement se cacher pour échapper aux hordes de morts-vivants. L'intérêt est donc plutôt dans le contexte et la mise en scène, et il faut bien avouer que les scénaristes ont été parfois très inspirés pour trouver les idées les plus loufoques, voire trop, notamment celles où des animaux zombies terrorisent les populations

(cf. https://www.senscritique.com/liste/les_animaux_zombiefies/392754).

2 l'énoncé

Le dieu du calembour foireux a décidé de faire son propre film de zombie en constatant qu'en verlan zombie devient bison. Le pitch : un cow-boy avec un très mauvais sens de l'orientation remonte une ligne de chemin de fer pour rejoindre le lieu de son prochain contrat. Malheureusement pour lui, un bison zombie suit la même voie en sens inverse. Pourquoi ? On s'en fout, car c'est un film de zombie. Le cow-boy pourra-t-il survivre à son voyage en se cachant dans les gares le long de la voie ? Telle est la question pas encore tranchée.

Pour tester différents scénarios, le dieu fait appel à une SSII, qui fait appel à vous pour simuler différentes situations. Celles-ci sont décrites par une chaîne de caractères de longueur N (variable selon les situations, mais $N \leq 10000$), composée de points, de caractères C pour les cachettes, d'un unique caractère H représentant l'humain, et d'un unique caractère Z représentant le zombie. Les ., C, H et Z peuvent se trouver n'importe où dans cette chaîne mais **jamais** au même endroit. Par exemple : ..H..C.C..Z.. ou encore ..C.H.Z.C. Pour simuler le déplacement des intervenants, on utilise un pas de temps plutôt qu'un déplacement en continu. A chaque pas de temps, on suppose que Z se déplace de 1 caractère vers la gauche, H de 1 caractère vers la droite ou 0 s'il se cache. L'objectif de H est d'atteindre la fin de chaîne qui représente l'arrivée. La simulation doit déterminer si H va :

- croiser le zombie sans être caché, et donc mourir,
- croiser le zombie en étant caché dans la cachette la plus proche possible de l'arrivée, et donc survivre,
- atteindre la fin de chaîne sans jamais avoir besoin de se cacher.

Important : les déplacements étant concurrents, on considère que l'humain peut se cacher s'il atteint une cachette exactement en même temps que le zombie. Il peut également atteindre la cachette avant et s'y cacher pendant plusieurs tours, c'est-à-dire le temps que le zombie atteigne lui-même la cachette. En revanche, si le zombie est juste après l'humain et que ce dernier bouge, alors il meurt forcément. Ces situations sont illustrées par les exemples suivants.

- **H.C.Z** : après 2 déplacements, H et Z arrivent au même moment sur C. H peut donc se cacher pendant tout un tour et survivre.
- **HC.Z** : après 1 déplacement, H arrive sur C et Z juste après. Le tour suivant, si H sort de la cachette, il rencontre Z et meurt. H doit donc se cacher pendant deux tours pour survivre.
- **H.CZ** : après 1 déplacement, Z arrive sur C, et H juste avant. Lors du déplacement suivant, H pourrait arriver sur C. Mais comme Z avance aussi, H n'a pas le temps de se cacher et meurt.

Pour tester les situations, votre programme doit lire sur l'entrée standard :

1. un entier M représentant le nombre de situations à tester
2. M chaînes de caractères au format décrit ci-dessus (avec ., C, H, Z), représentant chacune une situation.

Ensuite, votre programme doit déterminer pour chaque situation, le sort de l'humain et écrire sur la sortie standard :

- **dead** si l'humain meurt,
- **-1** s'il peut atteindre l'arrivée sans jamais avoir besoin de se cacher,
- **X** s'il peut atteindre l'arrivée en se cachant, **X** étant l'indice de la cachette, en les numérotant de gauche à droite et en partant de 1.

Le tableau 1 donne un exemple d'entrée et la sortie associée.

entrée	sortie
4	1
...H.C..CC..Z	2
....HC..CC..Z..	3
....HC..CC.....Z.....	dead
H.....C.Z..	

TABLE 1 – Exemple d'entrée et de sortie associée

Commentaires :

- situation 1 : H n'a pas le temps d'atteindre la cachette 2. Il ne peut donc qu'aller dans la 1 pour se cacher et attendre le temps que le zombie le dépasse.
- situation 2 : H arrive à la cachette 2 en même temps que Z. Il peut donc se cacher, puis poursuivre jusqu'à l'arrivée.
- situation 3 : H arrive à la cachette 3 un pas de temps avant Z. Il peut donc se cacher, attendre, puis atteindre l'arrivée.
- situation 4 : H n'a pas le temps d'arriver à la cachette 1 avant de croiser Z. Il meurt.

3 les ressources

Pour vous aider dans la réalisation du programme, vous trouverez sur <http://cours-info.iut-bm.univ-fcomte.fr> un article dans la section `hackathon` portant le même titre que l'exercice. Il contient un lien permettant de télécharger un canevas de code, ainsi que le fichier d'entrée donné ci-dessus.

Bien entendu, vous êtes libres d'utiliser ou non ce canevas, mais c'est un gain de temps que de s'en servir comme base.