SAE - Interface du moteur de jeu

Le moteur de jeu se compose de 6 exécutables, détaillés ci-dessous. Pour chacun, les requêtes sont données en ligne de commande (il n'y a pas de lecture de l'entrée standard), et les résultats sont donnés soit dans la valeur de sortie, soit en sortie standard.

Les exécutables s'appuient sur trois types de fichiers (pour les dictionnaires bruts et binaires, et les tableaux de fréquences de lettres), détaillés en fin de fichier.

Par défaut la valeur de sortie sera 0 en l'absence de problème, 1-255 en cas d'erreur. On utilisera les valeurs 1,2,... pour des erreurs courantes (mot incorrect, etc.), et 255, 254, ... pour des anomalies (fichier introuvable, paramètres manquants ou incorrects, etc.).

Exécutables attendus

`dictionnary_build`

```
$ dictionnary_build dico.txt dico.lex
```

Construit un arbre lexicographique à partir d'un dictionnaire (dico.txt), le sauvegarde dans un fichier binaire (dico.lex). Aucune sortie n'est attendue.

`dictionnary_lookup`

```
$ dictionnary_lookup dico.lex BONJOUR
[valeur de sortie = 0]
```

Recherche si un mot (BONJOUR) est dans un dictionnaire (dico.lex). Renvoie 0 si le mot est présent, 1 si c'est un préfixe valide d'un mot présent (par exemple, BONJ), 2 sinon. Aucune sortie n'est attendue

`grid_build`

```
$ grid_build frequences.txt 4 4
G A I R R U V E QU E O T A S M J
```

Crée et affiche une grille aléatoire de dimensions données (4 lignes, 4 colonnes) à partir d'un fichier de fréquences (frequences.txt). La grille est lue ligne par ligne, les cases sont séparées par des espaces.

`grid_path`

```
$ grid_path OUI 4 4 G A I R R U V E QU E O T A S M J
10 5 2
[valeur de sortie = 0]
```

Teste si un mot donné (OUI) est présent dans la grille (les paramètres sont hauteur, largeur), renvoie 0 si le mot est présent, 1 sinon. Si le mot est présent, on affiche le chemin emprunté (liste des indices des cases utilisées, séparés par des espaces). Aucune sortie si le mot est absent, par exemple:

```
$ grid_path TOIT 4 4 G A I R R U V E QU E O T A S M J
[valeur de sortie = 1]
```

solve

```
$ solve dico.lex 3 4 4 G A I R R U V E QU E O T A S M J
OUI VIE VIRE ...
```

Renvoie tous les mots valides de la grille présents dans le dictionnaire (dico.lex) séparés par des espaces, avec une taille minimum (3).

score

```
$ score OUI VIE
2
```

Reçoit une liste de mots en paramètres, et affiche un score associé. On suppose que tous les mots sont valides (présents dans la grille et le dictionnaire). Ces spécifications peuvent être adaptées en fonction de la règle de calcul du score: l'exécutable peut également recevoir l'adresse du dictionnaire de référence, les listes des autres joueurs, etc. On pourra avoir plusieurs exécutables correspondant à plusieurs choix de politique de scores.

Formats de fichiers

dictionnaire en texte brut (.txt)

Un fichier contenant tous les mots admis du dictionnaire (un par ligne). Les mots sont en majuscule, en utilisant uniquement les caractères A-Z (pas d'accent, de ponctuation, etc.). Ils peuvent éventuellement être suivis d'un espace et d'un commentaire, (par exemple pour indiquer une valeur utilisée dans le calcul de score). L'ordre alphabétique n'est pas nécessairement respecté. Une ligne commençant par un espace est ignorée.

```
dico.txt:
    AA
    AALENIEN
    AALENIENNE (éventuel commentaire ici)
    AALENIENNES
    AALENIENS
    AAS
    ABACA ## un autre commentaire
    ABACAS
    ABACOST
    ...
```

Fichier de fréquences (.txt)

Contient toutes les lettres ou paires de lettres pouvant apparaître dans une case de la grille (i.e. sur la face d'un dé), chacune suivie de sa fréquence (un entier positif).

```
frequences.txt:
    A 36
    B 5
    C 24
    ...
    P 15
    QU 26
    R 37
    ...
```

Dictionnaire au format binaire (.lex)

Ce dictionnaire contient l'équivalent d'un dictionnaire, mais organisé dans un fichier binaire afin d'être le plus efficace en mémoire. Le format doit respecter les contraintes suivantes:

- Le fichier commence par un en-tête, suivi d'un tableau de cellules

- L'entête contient (au moins) 4 champs:
  - la taille de l'en-tête (comme obtenu avec sizeof)
  - le nombre de mots dans le dictionnaire
  - le nombre de cellules dans le tableau
  - la taille de chaque cellule (comme obtenu avec sizeof)
- Chaque cellule contient (au moins) 3 champs, correspondant à une implémentation statique d'un arbre lexicographique

Vous êtes libres d'insérer des champs supplémentaire dans l'en-tête ou dans chaque cellule.

Ci-dessous un exemple indicatif (avec des valeurs choisies "à la main")

Bloc	Position	Valeur	Signification
En-tête	0	16	Taille de l'en-tête
	4	300	Nombre de mots
	8	1000	Nombre de cellules
	12	12	Taille de chaque cellule
Cellule 0	16	'A'	Element
	20	26	FirstChild
	24	25	nSibling
Cellule 1	28	'B'	Element
	32	37	FirstChild
	36	24	nSibling
Cellule 2	40	'C'	Element
	44	64	FirstChild
	48	24	nSibling
Cellule 3	52	'D'	Element
...	..	...	...
Cellule 999	12012	0	nSibling