

Interrogation 2 – Grille de mots croisés

Consignes

- Vous pouvez utiliser **toutes les notes écrites que vous voulez** ainsi que tous les programmes qui se trouvent dans votre dossier.
- A faire **exclusivement sous Linux**. Vous disposez de **2 heures**.
- Appelez votre classe MotsCroises et écrivez-là dans un dossier appelé interro2.
- Chaque fois qu'un point est terminé, copiez la classe dans le **casier** de votre professeur.
- Soyez attentifs, au **style**, à l'**indentation** et la **clarté** du code.

Le problème

Nous allons réaliser un petit programme d'aide à la résolution de mots croisés. Evidemment, en 2h de temps, nous ne pourrons écrire qu'une version très simplifiée.

Concrètement, il sera possible de remplir une grille mais aussi de chercher dans un dictionnaire tous les mots qui peuvent être placés à un endroit de la grille.

Entête

(1 point)

Ecrire une fonction qui affiche un entête comme celui-ci.

```
=====
Interrogation 2 - votre nom - votre groupe
=====
```

Comparer 2 mots

(3 points)

Ecrire une fonction qui reçoit un mot complet et un mot incomplet (comportant des lettres non spécifiées) et qui vérifie si le mot complet est un candidat valable pour le mot incomplet. Dans le mot incomplet, chaque lettre manquante sera indiquée par un espace.

Module estCandidat(motIncomplet, motComplet : Chaîne) → booléen

Exemples :

- estCandidat("ec le","ecole") retourne vrai
- estCandidat(" sc le","ecole") retourne faux

Note: La classe `String` offre la méthode `length()` pour connaître la taille d'une chaîne.

Lire une grille

(3 + 1 points)

Ecrire une fonction qui lit (sur l'entrée standard) une grille de mots croisés (N x M). La fonction lit d'abord le nombre de lignes et ensuite une série de chaînes représentant chacune une ligne de la grille. Les cases noires sont représentées par le caractère dièse ('#').

Voyez le fichier `/usr/local/java/evaluations/interro2/grille` pour un exemple.

Module lireGrille() → Tableau [N x M] de caractères

Note: La fonction `toCharArray()` de la classe `String` convertit une chaîne en un tableau de caractères. Nous vous demandons d'utiliser efficacement cette fonction pour répondre à la question.

Afin de vérifier la validité de cette fonction, écrivez une fonction qui affiche la grille.

Module afficherGrille(grille : Tableau [N x M] de caractères)

Donner les mots possibles

(4 points)

Ecrire une fonction qui trouve la liste des mots possibles pour compléter un mot donné. Vous utiliserez la fonction `dico()` de la classe `evaluations.interro2.Dico` pour obtenir un tableau de mots qui vont servir pour la recherche. La liste des mots trouvés est simplement affichée à l'écran.

Module listeMotsPossibles(motIncomplet : Chaîne)

Faire propositions

(4 points)

Ecrire une fonction qui fait des propositions pour un mot de la grille. Elle affiche la liste des mots pouvant se placer à la place indiquée.

Module propositions(grille: Tableau [N x M] de caractères,
ligne: Entier, colonne : Entier, horizontal : booléen)

où 'ligne' et 'colonne' désignent les coordonnées du début du mot et 'horizontal' indique si on considère un mot horizontal ou vertical. On peut supposer que les paramètres sont corrects.

Placer un mot

(2 points)

Ecrire une fonction qui place un mot dans la grille. On peut supposer que les paramètres sont corrects.

Module placerMot(grille: Tableau [N x M] de caractères,
mot : Chaîne,
ligne: Entier, colonne : Entier, horizontal : booléen)

Vérification des paramètres

(2 points)

Modifier les fonctions `propositions()` et `placerMot()` afin de vérifier la validité des paramètres.

- ligne et colonne désignent bien une case qui est un début de mot.
- le mot à exactement la bonne taille.