

Des chiffres et Des lettres



Des chiffres et des lettres est un jeu télévisé français créé en 1965 par Armand Jammot. Le but de ce projet est de créer une version numérique du jeu, puis de créer une intelligence artificielle permettant de trouver les meilleures solutions. Vous pourrez si vous souhaitez tester ce jeu aller sur le site France 3 pour accéder au jeu en ligne.

1. Rappel de quelques commandes

Pour réaliser ce TP, on pourra utiliser les différentes méthodes dédiées aux listes ou aux chaînes de caractères. L'étant une liste : `L.count(6)`, `L.remove(6)`, `L.append(9)`.

Python

```
# Modules
from random import randint # pour avoir accès à un tirage aléatoire d'entiers
import copy # avec la commande deepcopy. Copie de listes (pour éviter les alias).

#commandes : accès au fichier
f=open("dictionnaire.txt", 'r')
listeMot=f.readlines() # affecte dans une liste de lignes la totalité du fichier.
listeMot=f.readline() # permet de lire le fichier ligne par ligne

g=open("dictionnaire_trie.txt", "a")
g.write('atome')
```

2. Création du jeu

2.1. Le compte est bon

Le but de cette épreuve est d'obtenir un nombre (de 100 à 999) à partir d'opérations élémentaires (+, -, *, /) sur des entiers naturels, en partant de nombres tirés au hasard (de 1 à 10, 25, 50, 75 et 100). Lorsque l'émission n'était pas informatisée, le jeu comportait vingt-quatre plaques : les nombres de 1 à 10 présents en double exemplaire et les nombres 25, 50, 75 et 100 présents en un seul exemplaire.

Sont tirées alors 6 valeurs et le nombre à trouver. À défaut de trouver le compte exact, il faut tenter de s'en approcher le plus près possible.

Exemple

Nombres tirés : 3, 100, 8, 8, 10, 6

Résultat demandé : 683

Solution :

$$6 * 100 = 600$$

$$8 * 10 = 80$$

$$600 + 80 = 680$$

$$680 + 3 = 683$$

1. Créer une fonction appelée `tirageChiffres` sans argument et permettant de générer aléatoirement le tirage des chiffres et le nombre à trouver. Les 6 chiffres tirés ne sont pas remis.

Cette fonction doit renvoyer une liste contenant les 6 chiffres ainsi que le nombre à trouver.

Python

```
>>>C= tirageChiffres ()
>>>print (C)
[6, 9, 8, 8, 100, 6, 330]
```

2.2. Le mot le plus long

Dans cette épreuve, les deux candidats décident chacun leur tour s'ils souhaitent qu'une voyelle ou une consonne soit tirée au sort, jusqu'à obtenir 10 lettres. Le but est de trouver le plus long mot possible en utilisant les lettres qui ont été tirées. (Les mots composés, les féminins et pluriels, ainsi que les verbes à l'infinitif, participe passé et participe présent sont acceptés. En revanche, les verbes conjugués, les noms propres et les abréviations sont refusés.)

Exemple

Lettres tirées : B M A E R I L A N E

Solutions possibles :

REMANIABLE (10 lettres), AMIABLE (7 lettres), AILEE (5 lettres) ...

2. Créer une fonction appelée `tirageLettres` permettant de générer un tirage des lettres. Cette fonction demandera à l'utilisateur s'il souhaite une consonne ('c') ou une voyelle ('v'), et s'il rentre autre chose, on le préviendra par un message. Le tirage a maintenant lieu avec remise (une même lettre peut être tirée plusieurs fois).
3. Créer une fonction appelée `verification(mot)` permettant de vérifier que le mot trouvé est dans le dictionnaire. Un fichier texte nommé `dictionnaire_trie.txt` servira de dictionnaire. Attention : vérifier avant de commencer le contenu du dictionnaire en parcourant le fichier.

3. Création de l'intelligence artificielle

3.1. Le compte est bon

4. Créer 4 fonctions permettant d'effectuer les opérations élémentaires :

- ◇ `multiplication(a,b)` calculant $a*b$;
- ◇ `addition(a,b)` calculant $a+b$;
- ◇ `soustraction(a,b)` calculant $a-b$ si $a>b$, $b-a$ sinon;
- ◇ `division(a,b)` calculant a/b si a est un multiple de b , b/a si b est un multiple de a , et retournant 0 sinon.

5. Créer une fonction appelée `solucChiffres`, prenant en paramètre un tirage, permettant d'afficher si le compte est bon ou non. Si le compte est bon, la fonction affichera aussi le détail des opérations conduisant au bon compte.

Aide : on peut envisager l'algorithme suivant :

- ◇ on tire une opération au hasard parmi les 4 opérations élémentaires (on pourra affecter un chiffre aux différentes opérations : 0 pour l'addition, 1 pour la soustraction...).
- ◇ on tire deux nombres au hasard parmi les 6 chiffres de tirage. On élimine ces deux nombres de cette liste pour ne pas les réutiliser.
- ◇ on effectue l'opération et on stocke le résultat de l'opération dans la liste (mise à jour du "tirage" pour que ce résultat puisse être réutilisé ultérieurement).
- ◇ on recommence en tirant une nouvelle opération au hasard parmi les 4 opérations élémentaires, et en tirant un nombre au hasard parmi la liste contenant maintenant 5 nombres.
- ◇ ce processus jusqu'à ce que le compte soit trouvé, ou alors qu'il ne reste plus qu'un seul chiffre (qui constitue un résultat approché).

Il faut procéder ainsi de suite un très grand nombre de fois. On pourra limiter le nombre d'itérations à `n_max = 300000`. Si la boucle est réalisée `n_max` fois sans trouver le bon compte, on pourra conclure que le compte n'est pas bon.

6. Modifier la fonction précédente pour qu'elle affiche la meilleure solution approchée (et le détail des opérations) si aucune solution exacte n'a été trouvée.

```
Python
>>>> solucChiffres (C,30000)
        le compte est bon
        6+9=15
        8+6=14
        14+8=22
        22x15=330
        =330
```

```
Python
>>> solucChiffres (C,3000)
        le compte est approché
        9x8=72
        6x72=432
        8-6=2
        432-100=332
        =332
```

3.2. Le mot le plus long

Nous allons chercher à afficher automatiquement les 10 meilleures solutions du tirage de lettres.

Pour optimiser la recherche, on utilise toujours le dictionnaire `dictionnaire_trie.txt` qui ne possède pas de mots de plus de 10 lettres et dans lequel les mots sont classés par longueur (d'abord les mots de 10 lettres).

7. Créer une fonction appelée `solucLettres(tirage)`, prenant en paramètre un tirage, permettant d'afficher les 10 mots les plus longs trouvés.

8. S'il reste du temps, on pourra générer le dictionnaire trié : pour cela il faut mettre d'abord les mots de 10 lettres, puis ceux de 9 lettres, etc, et éliminer les mots de plus de 10 lettres. Créer une fonction appelée `nouvDico` permettant de générer le nouveau dictionnaire.