

ÉPREUVE D'ALGORITHMIQUE – E22

CONTRÔLE EN COURS DE FORMATION

DÉROULEMENT DE L'ÉPREUVE :

Cette épreuve de Contrôle en Cours de Formation (CCF) se déroule en trois étapes :

- **Étape 1 : Écrit (30 minutes)**

- ✓ Vous devez traiter la partie A du sujet par écrit.
- ✓ Les machines (ordinateur, calculatrice...) sont interdites pour cette partie.
- ✓ La partie A porte sur des algorithmes écrits en langage naturel. Les instructions écrites en langage Python seront néanmoins acceptées.
- ✓ Si vous terminez l'étape 1 (partie A) en moins de 30 minutes, vous pouvez commencer à préparer la partie B sur une feuille de brouillon, sans accès à l'ordinateur. L'accès à l'ordinateur n'est possible que lors de l'étape 2.

- **Étape 2 : Sur machine (30 minutes)**

- ✓ Vous devez traiter la partie B du sujet à l'aide d'un ordinateur.
- ✓ Devant l'ordinateur, vous ne pourrez plus écrire sur le sujet (tout stylo ou crayon est interdit).

- **Étape 3 : Oral (20 minutes)**

- ✓ Vous présentez votre solution algorithmique et son implémentation (10 minutes maximum)
- ✓ Vous participez à un entretien d'explicitation conduit par le professeur (10 minutes maximum)

Partie A - Durée : 30 minutes

Préparation sur table. L'utilisation d'une machine est interdite.

Toutes les réponses sont à rédiger sur le sujet dans les cadres prévus à cet effet.

Présentation du sujet

On souhaite programmer une version très simplifiée du jeu du Mastermind.

L'ordinateur va choisir une combinaison de 4 chiffres distincts aléatoirement choisis entre 0 et 7, et le joueur va devoir les deviner grâce aux indications fournies après chaque proposition.

Après chaque proposition, l'ordinateur affichera un message indiquant le nombre de chiffres corrects bien placés, et le nombre de chiffres corrects mal placés.

Le joueur dispose de 8 essais au maximum.

A la fin de la partie, l'ordinateur affiche suivant le cas « Gagné en ... coups » ou « Perdu, la bonne réponse était ... ».

Exemples

Si la combinaison secrète est 5423 et que le joueur propose 2463, l'ordinateur va indiquer deux chiffres bien placés (4 et 3) et un mal placé (2).

Si la combinaison secrète est 5216 et que le joueur propose 1264, l'ordinateur va indiquer un chiffre bien placé (2) et deux mal placés (1 et 6).

On donne ci-dessous l'algorithme d'une fonction.

Fonction transform(propos)

ARGUMENT propos : chaîne de caractères

TYPE RENVOYÉ tableau d'entiers à une dimension (longueur 4)

VARIABLES i : entier

 res : tableau d'entiers

DébutFonction

 res ← [0,0,0,0]

 Pour i allant de 0 à 3

 res[i] ← int(propos[i]) # int() convertit en nombre entier

 FinPour

 Renvoyer res

FinFonction

Question A-1

Indiquer le rôle de la fonction transform(). Quel résultat renvoie transform("1234") ?

La fonction `indic(combi, propos)` donnée ci-dessous prend pour paramètres d'entrée deux tableaux d'entiers à une dimension : `combi`, correspondant à la combinaison de 4 chiffres distincts à deviner, et `propos`, correspondant à la proposition faite par le joueur. Cette fonction renvoie un tableau d'entiers à une dimension, dont la première cellule indique le nombre de chiffres corrects bien placés, et la deuxième, le nombre de chiffres corrects mal placés.

Question A-2

Compléter l'algorithme de la fonction `indic(combi, propos)`.

Fonction `indic(combi, propos)`

ARGUMENTS `combi, propos` : tableaux d'entiers à une dimension

TYPE RENVOYÉ tableau d'entiers à une dimension

ROLE détermine le nombre de chiffres corrects bien placés et mal placés

VARIABLES `i, j, bPlace, mPlace` : entiers

`res` : tableau d'entiers à une dimension

DébutFonction

`bPlace` \leftarrow 0

`mPlace` \leftarrow 0

Pour `i` allant de 0 à 3 # parcours du tableau `propos`

Pour `j` allant de 0 à 3 # parcours du tableau `combi`

Si `propos[i]=combi[j]`

Si ...

`bPlace` \leftarrow ...

compléter

FinSi

FinSi

FinPour

FinPour

...

affectation du tableau `res` à renvoyer

Renvoyer `res`

FinFonction

On suppose que l'on dispose de la combinaison à deviner, affectée à la variable `combi`, de type tableau de 4 entiers à une dimension.

Question A-3

Écrire la portion de l'algorithme du programme principal permettant la saisie par le joueur de la combinaison proposée (à enregistrer dans un tableau de même type que `combi`), affichant le nombre de chiffres corrects bien placés, mal placés, puis proposant une nouvelle saisie de proposition jusqu'à ce que le jeu soit terminé, avec affichage sous la forme « Gagné en ... coups » ou « Perdu, la bonne réponse était ... ».

Une fonction `genere_combinaison()` permet de générer une combinaison aléatoire de 4 chiffres distincts entre 0 et 9.

Question A-4 (bonus)

Écrire l'algorithme de la fonction `genere_combinaison`. On suppose qu'on dispose d'une fonction `entier_aleatoire_entre(a, b)` renvoyant un entier aléatoire entre deux entiers `a` et `b`.

Partie B - Durée : 30 minutes

Préparation sur machine. Tout stylo ou crayon est interdit.

Votre fichier est à enregistrer sur la clé USB fournie sous le nom : `NomPrenom.py`

Question B-1

Implémenter en Python l'algorithme de la fonction `indic`.

Question B-2

Écrire le programme principal correspondant à l'algorithme écrit dans la partie A, en utilisant éventuellement la fonction `transform`.

Question B-3 (bonus)

Rajouter le tirage aléatoire de la combinaison.

`randint(a, b)` renvoie un entier aléatoire entre `a` et `b` inclus après **from** `random` **import** `randint`