LIFAPI – Partie A - Algorithmique

Contrôle Continu Terminal (Durée totale : 2h)
Lundi 18 décembre 2023

NOM :

……

PRENOM :

……

Numéro Etudiant :

……

Recommandations : *Les documents, calculatrice, téléphone portable sont interdits. La qualité de l'écriture et de la présentation seront prises en compte dans la note finale. Vous veillerez à* ***respecter*** *les notations et les règles d'écriture des algorithmes vues en cours et en TD. Un soin tout particulier devra être apporté à l'écriture des entêtes des différents sous-programmes.*

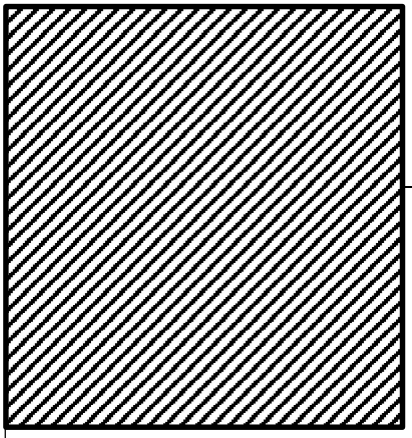
En mathématiques, l'algorithme de Kaprekar est un **algorithme qui transforme un nombre entier en un autre, de façon répétitive jusqu'à arriver à une valeur particulière**. Prenez un nombre à 4 chiffres : 3728 par exemple. Triez les 4 chiffres en ordre décroissant ➔ 8732. Triez les 4 chiffres en ordre croissant ➔ 2378. Soustrayez au premier nombre (8732) le deuxième nombre (2378). Ce qui donne : 8732 - 2378 = 6354

Avec le résultat obtenu (6354), reproduisez le raisonnement (algorithme de Kaprekar) ci-dessus et ainsi de suite. On obtiendra alors les opérations suivantes (depuis le début avec l'exemple 3728) :

8732 - 2378 = 6354
6543 - 3456 = 3087
8730 - 0378 = 8352
8532 - 2358 = 6174
7641 - 1467 = 6174

Quel que soit le nombre à 4 chiffres choisi, l'algorithme convergera toujours vers 6174 au bout de plusieurs itérations, sauf pour les 10 nombres comprenant quatre chiffres identiques : 0000, 1111, 2222, ..., 9999.

1- Ecrire l'algorithme d'une **fonction** `saisie_valeur` qui demande à l'utilisateur un entier **strictement supérieur à 1000, strictement inférieur à 9999** et différent de 1111 et 2222, 3333, 4444, 5555, 6666, 7777, et 8888 et retourne la valeur choisie. La saisie sera recommencée tant que la valeur proposée ne respecte pas ces contraintes.



2- Ecrire l'algorithme du sous-programme `int2tab` qui prend en paramètre un entier `n` strictement inférieur à 10000 (sur 4 chiffres) et qui construit et "retourne" un tableau `tab` de 4 entiers contenant chacun des chiffres du nombre `n`. Exemple si `n= 2345` on construira `tab`

2	3	4	5

3- Ecrire l'algorithme d'une **fonction** `tab2int` qui à partir d'un tableau de 4 entiers positifs strictement inférieurs à 10 calcule et retourne l'entier correspondant.
Exemple avec `tab`

2	3	4	5

on renverra `2345 (= 2*1000+3*100*4*10+5)`

4- Ecrire l'algorithme d'un sous-programme `renverse_tab` qui à partir d'un tableau de 4 entiers construit et "retourne" le tableau inversé. Exemple | 2 | 3 | 4 | 5 | donnera | 5 | 4 | 3 | 2 |

5- On dispose d'une procédure `trie_tab(int T_init[4], int T_trie[4])` qui trie le tableau `T_init` dans le tableau `T_trie`.
Ecrire l'algorithme d'une **fonction** `kaprekar` qui, à partir d'un entier à 4 chiffres (respectant les contraintes de la question1-) passé en paramètre, calcule et retourne le nombre d'itérations nécessaires pour arriver à `6174` en appliquant l'algorithme décrit en introduction. On utilisera `tab2int`, `int2tab`, `trie_tab` et `renverse_tab` à chacune des étapes.

6- Ecrire l'algorithme du programme principal qui demande à l'utilisateur un entier entre `1001` et `9998`, et affiche le nombre d'itérations nécessaires à l'algorithme de `kaprekar` pour converger vers `6174`.