
BTS SIO

SOUS-ÉPREUVE E22

ALGORITHMIQUE APPLIQUÉE

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. Pour cette partie, l'ordinateur est interdit mais la calculatrice est autorisée.

Vous inscrirez vos réponses dans le document réponse à la fin du sujet.

Les algorithmes à écrire peuvent être rédigés en **langage naturel** ou en PYTHON mais ni en C# ni en VB.NET.

À la fin de l'étape 1, votre document réponse doit être remis à la personne surveillant l'épreuve. Vous garderez le sujet.

– **Étape 2 : sur machine (30 minutes)**

Vous devez traiter la partie B du sujet à l'aide d'un ordinateur. Le langage utilisé est celui travaillé dans l'année, à savoir PYTHON. Vous sauvegarderez votre travail sur la clé USB fournie.

La durée totale pour effectuer les deux premières étapes est exactement d'une heure.

– **Étape 3 : oral (20 minutes au maximum)**

Cette partie se déroule en deux temps. Tout d'abord, vous disposez de 10 minutes pour présenter votre travail de l'étape 2 puis, au cours des 10 minutes suivantes, un entretien permet de préciser votre démarche.

À la fin de l'épreuve le sujet devra être rendu à l'examineur

Nombres Harshad

Étape 1

1. On considère le programme suivant

Code Python

```
chaine = input("Entrez un entier positif :")
longueur = len(chaine)
somme = 0
for i in range(longueur):
    chiffre = int(chaine[i])
    somme = somme + chiffre
print(somme)
```

- a. Expliquer le rôle de la ligne `chiffre = int(chaine[i])` du programme.
 - b. Faire sur la copie à rendre un tableau d'étapes de ce programme pour $n = 12$, $n = 216$ et $n = 8!$ (qui vaut $1 \times 2 \times 3 \times \dots \times 8$).
2. Un nombre Harshad est un entier positif qui est divisible par la somme des chiffres de son écriture en base 10.
Par exemple 48 est un nombre Harshad : la somme de ses chiffres vaut $4 + 8 = 12$ et $48 = 4 \times 12$ donc 48 est divisible par 12.
 - a. Vérifier que 12, 216 et $8!$ sont des nombres Harshad.
 - b. Montrer que 19, 155 et 2416 ne sont pas des nombres Harshad.
 3. Les premiers nombres Harshad avec plus d'un chiffre en base 10 sont :

10, 12, 18, 20, 21, 24, 27, 30, 36, 40, 42, 45, 48, 50, 54, 60, 63, 70, 72, 80, 81, 84, 90, 100, 102, 108, 110, 111, 112, 114, 117, 120, 126, 132, 133, 135, 140, 144, 150, ...

On peut constater que les entiers 110, 111, 112 sont trois nombres Harshad consécutifs.

On se propose de chercher parmi les nombres Harshad inférieurs à 1000, s'il existe d'autres séries de trois entiers consécutifs et de les afficher.

Pour cela, on suppose que l'on a mis dans une liste nommée `H` les nombres Harshad inférieurs ou égaux à 1000. On note `len(H)` le nombre de valeurs de cette liste (longueur de la liste).

Pour tester s'il y a trois nombres consécutifs dans ce tableau on écrit le programme suivant :

Code Python

```
for i in range(len(H)):
    if H[i] - H[i - 1] == 1 or H[i + 1] - H[i] == 1:
        print(H[i - 1], ', ', H[i], ' et ', H[i + 1], ' sont 3 nombres
        ↪ Harshad consécutifs.')
```

Ce programme comporte deux erreurs. Corrigez-les.

Étape 2

1. Sur la clé qui vous a été donnée, ouvrir le fichier **somme.py**. Celui-ci contient le programme de l'étape 1.
2. Transformez ce programme en une fonction nommée **somme_chiffres** qui
 - en entrée prend un **int** positif **n**;
 - renvoie la somme des chiffres de **n**.
3. Écrire une fonction **liste_harshad** utilisant la fonction précédente, qui
 - en entrée prend un **int** positif **n**;
 - renvoie la liste des nombres Harshad inférieurs ou égaux à **n**.
4. Construire une fonction **trois_harshad** qui utilise la fonction précédente et
 - en entrée prend un **int** positif **n**;
 - renvoie la liste des séries de trois nombres Harshad consécutifs inférieurs ou égaux à **n**.
5. À l'aide de la fonction précédente, construire la fonction **trois_harshad_bornes** qui
 - en entrée prend en argument deux **int** positifs **a** et **b**;
 - renvoie la liste des séries de trois nombres Harshad consécutifs compris entre **a** et **b** (inclus).
6. Quelles sont les séries de trois nombres Harshad consécutifs compris entre 1000 et 5000 ?
7. Construire une nouvelle fonction **quatre_harshad_bornes** pour déterminer les séries de quatre nombres Harshad consécutifs entre deux entiers naturels **a** et **b** donnés.
8. Exécuter cette fonction
 - pour **a=10** et **b=5000**;

– pour $a=10000$ et $b=15000$;

9. BONUS : Faire afficher la liste des nombres Harshad dont le chiffre des dizaines est 7. Faire afficher la liste des nombres Harshad non divisibles par 3.