
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.

Étape 1

On considère un entier $n \geq 1$ et on lui applique le processus suivant :

Processus

- si n est pair, on le divise par 2;
- sinon, on le multiplie par 3 et on ajoute 1.

Par exemple, si on choisit 5, puisqu'il est impair, on le multiplie par 3 et on ajoute 1, ce qui fait 16.

Le but de cette épreuve est d'étudier des propriétés de ce processus lorsqu'on le répète plusieurs fois à la suite.

Question 1

- Répéter ce processus huit fois en prenant 5 comme valeur de départ en indiquant les résultats successifs.
Comment cela va-t-il se poursuivre ?
- Répéter 4 fois ce processus en prenant 13 et expliquer comment on peut continuer un peu « sans se fatiguer ».

Le résultat suivant n'est pas (encore) démontré :

Conjecture

Quel que soit l'entier positif n que l'on prend au départ, en répétant ce processus on atteint *inmanquablement* la valeur 1, après quoi le cycle (4, 2, 1) se répète indéfiniment.

On aimerait écrire une fonction `temps_de_vol` qui

- en entrée prend un entier n
- renvoie le nombre de répétitions du processus nécessaire pour obtenir la valeur 1.

Question 2

- Que devra renvoyer `temps_de_vol(5)` ?
- Que devra renvoyer `temps_de_vol(13)` ?
- Compléter sur votre copie la ligne de définition de la fonction avec les indications

de type nécessaires :

Python

```
def temps_de_vol(...) -> ... :
```

Question 3

Compléter sur votre copie la fonction suivante.

Python

```
resultat = 0
while .....:
    if .....:
        n = .....
    else:
        n = .....
    resultat = .....
return .....
```

On appelle *hauteur de vol* de n le plus grand nombre qu'on obtient quand on répète le processus jusqu'à obtenir 1.

Question 4

- Quelle est la hauteur de vol de 5? Quelle est celle de 13?
- Écrire en PYTHON la fonction `hauteur_de_vol`

Étape 2

Ouvrir le fichier `syracuse.py`.

Celui-ci contient une implémentation de la fonction `temps_de_vol` de l'étape 1.

Question 5

Implémenter la fonction `hauteur_de_vol` de l'étape 1

On dit qu'un nombre vole bien lorsque son temps de vol est plus grand ou égal à lui-même. Par exemple, 7 vole bien car son vol est

(22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1)

et sa longueur (16), est plus grande que 7.

Question 6

Coder la fonction `vole_bien` qui

- en entrée prend un entier n positif;
- renvoie `True` si n vole bien et `False` sinon.