

TP 1 informatique

BCPST 2 2019-2020

Révisions

V.Vong

Points abordés

- Opérateurs logiques, comparaisons.
 - Structures conditionnelles.
 - Boucles for et while.
 - Fonctions.
-

Exercice 1. Soit $a \in \mathbb{N}$. On définit la suite $(a_n)_{n \in \mathbb{N}}$ par :

$$a_0 = a, \forall n \in \mathbb{N}, a_{n+1} = \begin{cases} \sqrt{a_n} & \text{si } \exists p \in \mathbb{N}, p^2 = a_n \\ a_n + 3 & \text{sinon} \end{cases}$$

1. Écrire une fonction `suite(a,n)` qui retourne la valeur de a_n .
2. Écrire une fonction `suiteL(a,n)` qui retourne la liste $[a_0, a_1, \dots, a_n]$. Attention, on veillera à ce que chaque terme de la suite ne soit calculée qu'une seule fois.
3. Écrire une fonction `boucle(a,N)` qui renvoie `True` si renvoie `True` s'il existe $(p, q) \in \{0, \dots, N\}^2, p \neq q$ tel que $a_p = a_q$ et `False` sinon.
4. Écrire une fonction `premierCarreP(a)` qui détermine le premier n tel qu'il existe $p \in \mathbb{N}$ tel que $p^2 = a_n$.

Exercice 2. On considère le pseudo-code suivant :

```
fonctionmystere(x) :  
    P=1  
    n=1  
    tant que P<x faire  
        n=n+1  
        P=P*n  
    retourner n
```

1. Quelle est la valeur de `fonctionmystere(43)` ?
2. À quoi correspondent les valeurs P et n ?
3. Écrire en Python une fonction correspondant au pseudo-code précédent.

Exercice 3. Un client décide d'emprunter à la banque un montant de 150000 euros. La banque accepte de lui faire un crédit avec un taux annuel de 1,5 pour cent. Il effectue un remboursement de R euros par an. Écrire un programme déterminant le nombre d'années nécessaires pour finir le remboursement.

Exercice 4. On rappelle le jeu de la taupe : dans une borne d'arcade, se trouve un certain nombre de trous. Dans l'un d'entre eux, sort une taupe. Il faut alors à l'aide d'un marteau taper celle-ci.

Pour modéliser ce jeu on procède de la façon suivante : la borne d'arcade est modélisée par une liste de taille n contenant des 0 (symbolisant un trou) et un 1 (qui représente la taupe). On suppose que la taupe se déplace au hasard de manière uniforme. On modélise les coups du joueur, par des nombres de 0 à $n - 1$, correspondant aux indices de la liste. Dans cette version informatique, on comptera le nombre de fois qu'on a touché la taupe après k coups.

Écrire une fonction qui permet de jouer à la version donnée dans la modélisation.

Exercice 5. 1. Écrire une fonction `matrice_aleatoire(n)` qui retourne une matrice de taille $n \times n$ où les coefficients de la matrice sont des entiers compris entre 0 et 6 choisis au hasard.

2. Écrire une fonction `occurrence(M)` qui prend en argument une matrice carrée et qui retourne une liste L où $L[i]$ est le nombre de i dans la matrice. On restreindra au cas où i est compris entre 0 et 6.

Exercice 6. On considère le programme suivant :

```
a=1
def toto() :
    global a
    a=49
toto()
```

1. Déterminer la valeur de a à l'issue du programme.

2. Écrire un programme qui modifie la variable $L = [1, 2, 3]$ en inversant 1 et 2.

Remarque : en général, on évite d'utiliser les variables globales. Mais celles-ci peuvent être pratique dans certains cas.

Exercice 7. Le jeu du FizzBuzz consiste à énumérer les entiers de 1 à n en prononçant "fizz" si le nombre est un multiple de 3, "buzz" si le nombre est un multiple de 5. Dans le cas où le nombre est un multiple de 3 et 5 on prononce alors "fizzbuzz". Par exemple, pour les entiers de 1 à 3, on obtient : un, deux, fizz. Pour l'entier 5 on prononce "buzz" et pour l'entier 15, on prononce "fizzbuzz".

Écrire un fonction `fizzbuzz(n)` qui affiche tous les entiers de 1 à n en respectant les règles du fizzbuzz.

Exercice 8. On considère la suite définie par :

$$u_0 = 1, \forall n \in \mathbb{N}, u_{n+1} = \sum_{k=0}^n u_k u_{n-k}.$$

Écrire une fonction `suite(n)` qui prend en argument un entier n et qui renvoie la liste $[u_0, u_1, u_2, \dots, u_{n-1}]$.

Exercice 9. Étant donné un sous-ensemble fini A de \mathbb{N} , on définit $mex(A)$ par le plus petit entier positif qui n'est pas un élément de A . Écrire une fonction `mex(A)` qui prend en argument une liste d'entiers A et qui renvoie $mex(A)$.