

TP 5 informatique

BCPST 1 2019-2020

Fonctions

- Exercice 1.**
1. Écrire une fonction `factorielle(n)` qui retourne la valeur de $n!$. (On rappelle que $n! = 1 \cdot 2 \cdot \dots \cdot n$ et $0! = 1$)
 2. Calculer $30!$.
 3. En déduire une fonction `binomial(n,k)` qui retourne la valeur de $\binom{n}{k}$. (On rappelle que $\binom{n}{k} = \frac{n!}{k!(n-k)!}$)
 4. Calculer $\binom{25}{12}$.

- Exercice 2.**
1. Écrire une fonction `suite(n,a)` qui renvoie u_n où la suite $(u_n)_{n \in \mathbb{N}}$ est définie par :

$$u_0 = a, \forall n \in \mathbb{N}, u_{n+1} = 2u_n - 3.$$

2. Écrire une fonction `somme(n)` qui renvoie la valeur $\sum_{i=1}^n \left(\sum_{j=1}^{n^2} \frac{i(j+1)}{2} \right)$.

- Exercice 3.** Soit $a \in \mathbb{R}$. On définit la suite $(u_n)_{n \in \mathbb{N}}$ par :

$$u_0 = a, \forall n \in \mathbb{N}, u_{n+1} = u_n^3 - 6u_n + 2$$

1. Écrire une fonction `suiteU(a,n)` qui renvoie la valeur u_n .
2. En déduire une fonction `borneRangN(a,n,m,M)` qui renvoie `True` si pour tout $k \in \{0, \dots, n\}, m \leq u_k \leq M$ et `False` sinon.

Exercice 4. On rappelle que l'opérateur `+` sur les chaînes de caractères permet de concaténer (mettre bout à bout) deux chaînes de caractères. Par exemple, si on effectue `chaine='chat'+'souris'`, la variable `chaine` contient alors `'chatsouris'`. De plus, pour accéder au caractère en position i , on peut écrire `chaine[i]`. Dans l'exemple précédent, `chaine[4]` a pour valeur "s".

1. Écrire une fonction `retourne(chaine)` qui renvoie la chaîne lu de droite à gauche.
2. En déduire une fonction `palindrome(chaine)` qui renvoie `True` si `chaine` est un palindrome et `False` sinon.

Exercice 5. On dit que deux mots sont anagrammes si on peut obtenir le deuxième mot en échangeant des lettres du premier. Ainsi, on peut remarquer que deux mots sont anagrammes si et seulement s'ils ont les mêmes lettres avec les mêmes nombres d'occurrences (nombre de fois qu'apparaît une lettre) et qu'ils ont la même longueur.

1. Écrire une fonction `Occurrence(chaine,lettre)` qui prend en argument une chaîne de caractères `chaine` et une lettre `lettre` et qui retourne le nombre de fois où `lettre` apparaît dans `chaine`.
2. En déduire une fonction `anagramme(chaine1,chaine2)` qui renvoie `True` si ces deux chaînes sont anagrammes et `False` sinon. On se restreindra aux mots ne contenant que des lettres minuscules de l'alphabet et sans accent. On pourra commencer la fonction par :

```
def anagramme(chaine1,chaine2) :  
    alphabet="abcdefghijklmnopqrstuvwxyztz"  
    if len(chaine1)==len(chaine2) :  
        for j in range(len(alphabet)) :
```

Exercice 6. *

Un habitué du jardin du Luxembourg nourrit régulièrement une horde de pigeons. On sait que les pigeons arrivent toujours de la manière suivante :

- la première minute, un pigeon se présente,
- la deuxième minute, deux pigeons le rejoignent,
- la troisième minute, trois pigeons intègrent le groupe,
- et ainsi de suite.

Sachant qu'un pigeon picore une portion de miettes de pain en une minute, qu'il n'est jamais rassasié, et que les premiers pigeons sont les premiers servis, écrire une fonction `nourrirPigeon(NbrePortions)` qui prend en argument un entier qui correspond aux nombres de portions de miettes de pain, et qui renvoie le nombre de pigeons ayant été nourri au moins une fois.