

INF1 : Algorithmique et Programmation

Cours 14 : Révisions

Domitile Lourdeaux

Université de technologie de Compiègne

Printemps 2024



utc
Université de Technologie
Compiègne

- 1 Appartenance d'un élément dans un tableau
- 2 Réversivité
- 3 Fichiers
- 4 Divers

① Appartenance d'un élément dans un tableau

② Réversivité

③ Fichiers

④ Divers

Test d'appartenance d'un élément dans un tableau

```
def appartient(element, t):
    trouve = False
    i = 0
    while i < len(t) and not trouve :
        if t[i] == element:
            trouve = True
        else
            i += 1
    return trouve
```

Test d'appartenance d'un élément dans un tableau

Autre version discutable

```
def appartient(element, t):  
    for e in t:  
        if e == element:  
            return True  
    return False
```

Test d'appartenance en fonction d'une clé

Recherche d'une personne suivant son nom dans un tableau de personnes, chaque personne étant définie par un dictionnaire

```
def appartient(nom, personnes):
    trouve = False
    i = 0
    while i < len(personnes) and not trouve :
        if personnes[i]["nom"] == nom:
            trouve = True
        else:
            i += 1
    return trouve
```

- ① Appartenance d'un élément dans un tableau
- ② Récurtivité
- ③ Fichiers
- ④ Divers

Final P18

Multiplication

Un principe connu depuis fort longtemps permet d'effectuer la multiplication de deux entiers naturels a et b de la façon suivante

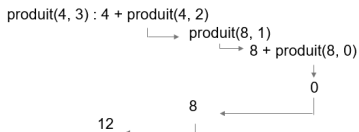
- si b vaut 0 alors : $a * b$ est égal à 0
 - si b est impair alors : $a * b$ est égal à $a + a * (b - 1)$
 - si b est pair alors : $a * b$ est égal à $(a + a) * (b/2)$
- 1 Ecrire une fonction récursive nommée produit qui applique ce principe pour multiplier deux nombres entiers a et b
 - 2 Faire une simulation avec $a = 4$, $b = 3$

Final P18

Multiplication

```
def produit(a, b):  
    if b == 0 :  
        result = 0  
    elif b % 2 == 0 :  
        result = produit(a + a, b // 2)  
    else :  
        result = a + produit(a, b - 1)  
    return result  
print(produit(4,3))
```

Simulation



Final P18

Triangle de Pascal

1						
1	1					
1	2	1				
1	3	3	1			
1	4	6	4	1		
1	5	10	10	5	1	
1	6	15	20	15	6	1

Final P18

Triangle de Pascal

1						
1	1					
1		1				
1			1			
1				1		
1					1	
1						1

1						
1	1					
1	2	1				
1			1			
1				1		
1					1	
1						1

Final P18

Triangle de Pascal

Pour calculer un terme de la ligne suivante, on prend le terme juste au-dessus, et on lui additionne celui qui est juste avant (0 s'il n'y a rien).

```
def triangle_pascal(ligne, colonne):  
    if colonne == 0 or colonne == ligne :  
        return 1  
    else :  
        return triangle_pascal(ligne - 1, colonne) + triangle_pascal(ligne - 1, colonne - 1)  
  
def affiche_triangle(n):  
    for ligne in range(n + 1) :  
        for colonne in range(ligne + 1) :  
            print(triangle_pascal(ligne, colonne), end = ' ')  
        print()  
affiche_triangle(7)
```

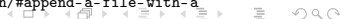
- ① Appartenance d'un élément dans un tableau
- ② Récurtivité
- ③ Fichiers
- ④ Divers

Fichiers

- 1 **Ouverture** du fichier et création d'un "objet fichier"
 - Lecture : 'r'
 - `file = open('my_file.txt', 'r')`
 - Ecriture 'w'
 - `file = open('my_file.txt', 'w')`
 - Le fichier est créé s'il n'existe pas et écrasé sinon
 - Ajout : 'a'
 - `file = open('my_file.txt', 'a')`
 - Les enregistrements sont ajoutés à la fin du fichier
 - Ajouter b pour les fichiers binaires : rb, wb, ab
- 2 Instruction(s) de **lecture** (ou itération sur chaque enregistrement du fichier) ou d'**écriture**
- 3 **Fermeture du fichier**
 - `file.close()`

Pour plus de détail voir :

<https://mkyong.com/python/python-difference-between-r-w-and-a-in-open/#append-a-file-with-a>



Lecture ligne à ligne en Python

Fin de fichier

```
ligne = ""
```

Lecture avec la méthode readline()

```
infile = open("poeme.txt", 'r')
ligne = infile.readline()
while ligne != "":
    print(ligne)
    ligne = infile.readline()
infile.close()
```

Test

```
C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent. Où le soleil de la montagne fière
```

Pour enlever `\n` :

```
ligne = ligne.strip()
Ou
ligne = ligne[:-1]
```

Chaque ligne se termine par `\n`.
=> une ligne vide après chaque ligne.

Lecture ligne à ligne en Python

Un objet fichier est un **"itérable"** en python. On utilisera donc de préférence la méthode suivante :

```
infile = open("poeme.txt", 'r')
for ligne in infile :
    print(ligne)
infile.close()
```

Test d'existence

Il est toujours préférable de tester l'existence du fichier en lecture

```
try :  
    infile = open("poeme.txt", 'r')  
    for ligne in infile :  
        print(ligne)  
    infile.close()  
except IOError :  
    print("Erreur : le fichier n'existe pas")
```

Écriture dans un fichier

- **Exemple**
 - Saisie d'un texte par l'utilisateur et écriture dans un fichier
- **Problème** : convention d'arrêt pour l'utilisateur
 - Ligne vide par exemple
- **Algorithme**

```
infile = open("poeme.txt", 'w')
ligne = " "
while ligne != "" :
    ligne = input("Ligne : ")
    infile.write(f'{ligne}\n')
infile.close()
```

N.B.

- Si pas de retour à la ligne, met tout bout à bout
- Si le texte contient une ligne vide, il faut une autre convention

Écriture dans un fichier

```
t = ['Paris', 'Marseille', 'Lyon', 'Toulouse', 'Nice']
outputfile = open("villes.txt", 'w')
for element in t :
    outputfile.write(f'{element}\n')
outputfile.close()
```

Contenu après écriture

Paris

Marseille

Lyon

Toulouse

Nice

Gestion des retours à la ligne

```
inputfile = open("villes.txt", 'r')
for ligne in inputfile :
    print(ligne)
inputfile.close()
```

Contenu après écriture

Paris

Marseille

Lyon

Toulouse

Nice

Gestion des retours à la ligne

```
inputfile = open("villes.txt", 'r')
for ligne in inputfile :
    ligne = ligne.strip()
    print(ligne)
inputfile.close()
```

Contenu après écriture

Paris

Marseille

Lyon

Toulouse

Nice

A savoir sur strip

- Par défaut enlève le retour à la ligne
- `s.strip("e")` enlève le "e" final si se termine par "e"

Ajout dans un fichier

```
inputfile = open("poeme.txt", 'a')  
inputfile.write("Rimbaud")  
inputfile.close()
```

- ① Appartenance d'un élément dans un tableau
- ② Récurtivité
- ③ Fichiers
- ④ Divers

Méthodes utiles, à revoir et à utiliser si nécessaire

split()

- `s = "Vivement les finaux terminés"`
- `t = s.split()`
- `print(t) : ["Vivement", "les", "finaux", "terminés"]`

Questions...