

RECURSIVITE

1. Introduction

2. Fonction récursive

3. Exemples :

4. Application avec le module Turtle :

5. Exercices : Activité-3

5.1 Calcul de C_n^p .

Pour calculer les coefficients binomiaux, on utilise la formule du triangle de Pascal:

$$C_n^p = C_{n-1}^p + C_{n-1}^{p-1}$$

On obtient successivement les lignes du triangle de Pascal

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

.....

a. Écrire un programme qui calcule successivement les lignes du triangle de Pascal :
lignesPascal(n) renvoie la liste des coefficient de la n-ième ligne. On fera un programme itératif.

Exemple : lignesPascal(4) renvoie [1,4,6,4,1]

En déduire un calcul de C_n^p dans le cas où $n \geq p$.

b. calculs récursifs mais avec **mémoire**

Écrire un programme qui à l'aide d'un dictionnaire et d'un programme récursif effectue le calcul des coefficients binomiaux C_n^p dans le cas où $n \geq p$. Les clés du dictionnaire seront des tuples(n,p) et les valeurs seront les C_n^p correspondants.

5.2 listes composées « irrégulières »

On considère des listes du type suivant : `ml=[[1,2],3,4,5],[[6,7]],8,9,[10,11,12],13,[14]]` c'est à dire des listes qui peuvent contenir des listes, pouvant elles-mêmes contenir des listes et ce à n'importe quel niveau.

a. nombre d'éléments d'une liste

On veut compter le nombre total d'éléments « simples » de telles listes. Écrire la fonction `compte`.

```
>>> compte(ml)
14
```

b. faire la somme des éléments d'une liste

Écrire la fonction `somme` qui renvoie la somme des éléments « simples » d'une telle liste.

```
>>> somme(ml)
105
```

c. lister les éléments : écrire une fonction qui renvoie la liste des éléments « simples » d'une telle liste .

```
>>> lister(ml)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

d. Recherche d'appartenance : écrire une fonction qui renvoie « true » si un élément appartient à une liste (à tous niveaux) et « False » si l'élément n'appartient pas à cette liste.

```
>>> recherche(1,ml)
True
>>> recherche(20,ml)
False
```

5.3 Binaire

Écrire une fonction récursive « bin » qui affiche la décomposition en binaire d'un entier passé en paramètre.

```
>>> bin(37)
0
1
0
0
1
0
1
```