

## TD2 - Algorithmique

### Exercice 1 : Coefficients binomiaux

On souhaite écrire une fonction permettant de calculer les coefficients binomiaux en s'appuyant simplement sur la formule de Pascal.

1. Écrire une fonction `somme_Pascal(L)` qui prend en entrée une liste (non vide) d'entiers `L` et qui renvoie la liste obtenue en additionnant deux termes consécutifs de `L` selon le modèle suivant :

Pour  $L = [a_0, a_1, a_2, \dots, a_n]$  la fonction devra renvoyer  $[a_0, a_0 + a_1, a_1 + a_2, \dots, a_{n-1} + a_n, a_n]$ .

```
>>> somme_Pascal([2, 5])
[2, 7, 5]
>>> somme_Pascal([6])
[6, 6]
>>> somme_Pascal([5, 8, 1, 6, 4])
[5, 13, 9, 7, 10, 4]
```

2. En déduire une fonction `coef_binome(k, n)` qui renvoie le nombre de combinaisons  $\binom{n}{k}$ .

```
>>> coef_binome(3, 5)
10
>>> coef_binome(33, 55)
1300853625660225
```

### Exercice 2 : Liste des diviseurs

Dans le cours, on dispose de la propriété suivante :

Si  $k \leq k'$  sont deux diviseurs associés d'un entier naturel non nul  $a$  alors  $k^2 \leq a$ .

Ainsi, pour déterminer l'ensemble des diviseurs positifs  $\mathcal{D}_{\mathbb{N}}(a)$  d'un nombre  $a$ , on teste les entiers les uns après les autres en commençant par 1, on note chaque diviseur obtenu avec son diviseur associé et on s'arrête lorsque le carré du nombre testé dépasse  $a$ . Voici deux exemples d'exécutions de cet algorithme.

Pour  $n = 12$  :

$1^2 = 1 \leq 12,$	1 divise 12 et son diviseur associé est 12
$2^2 = 4 \leq 12,$	2 divise 12 et son diviseur associé est 6
$3^2 = 9 \leq 12,$	3 divise 12 et son diviseur associé est 4
$4^2 = 16 > 12,$	c'est fini.

$$\mathcal{D}_{\mathbb{N}}(12) = \{1, 12, 2, 6, 3, 4\}$$

Pour  $n = 25$  :

$1^2 = 1 \leq 25$ ,	1 divise 25 et son diviseur associé est 25
$2^2 = 4 \leq 25$ ,	2 ne divise pas 25
$3^2 = 9 \leq 25$ ,	3 ne divise pas 25
$4^2 = 16 \leq 25$ ,	4 ne divise pas 25
$5^2 = 25 \leq 25$ ,	5 divise 25 et son diviseur associé est 5
$6^2 = 36 > 25$ ,	c'est fini.

$$\mathcal{D}_{\mathbb{N}}(25) = \{1, 5, 25\}$$

On rappelle que le quotient de la division euclidienne  $a$  par  $b$  s'obtient par  $a//b$  et le reste par  $a\%b$  ainsi pour tester si un nombre  $k$  divise un nombre  $n$ , on utilisera simplement la condition  $n\%k == 0$ .

Écrire une fonction `ensemble_diviseurs(n)` qui prend un entier  $n$  en paramètre et renvoie une liste qui contient les diviseurs de  $n$ .

```
>>> ensemble_diviseurs(25)
[1, 25, 5]
>>> ensemble_diviseurs(12)
[1, 12, 2, 6, 3, 4]
```

### Exercice 3 : Division euclidienne, première version

On souhaite effectuer la division euclidienne d'un entier naturel  $a$  par un entier naturel non nul  $b$ . Un algorithme simple consiste à retirer  $b$  à  $a$  tant que le résultat reste plus grand que  $b$ . Le nombre de soustractions effectuées donne le quotient et le dernier résultat d'une soustraction donne le reste.

Écrire une fonction `div_euclidienne1(a, b)` qui met en œuvre cet algorithme.

```
>>> div_euclidienne1(25, 4)
(6, 1)
>>> div_euclidienne1(248, 43)
(5, 33)
```

### Exercice 4 : Division euclidienne, deuxième version

Avec la fonction obtenu précédemment tester la division d'un nombre de 9, 10 puis 11 chiffres (le dividende) par un diviseur à deux chiffres. Demander ensuite à Python de faire pareil :

```
>>> div_euclidienne1(2485555555, 43)
(57803617, 24)
>>> 2485555555 // 43, 2485555555 % 43
(57803617, 24)
>>> div_euclidienne1(24855555555, 43)
(578036175, 30)
>>> 24855555555 // 43, 24855555555 % 43
(578036175, 30)
```

Que constatez-vous ?

En vous inspirant de la méthode de division « à la main », proposer un algorithme efficace.