

On se propose de coder l'addition de deux entiers naturels connus par leurs représentations en binaire sur N bits où N est une variable globale du programme.

Ouvrir le fichier `additionneur-binaire-eleve.py` qu'on complétera au fil de l'exercice.

1. Dans cette question, on fixe $N = 1$ bit.

- a. Compléter la table d'addition ci-dessous des entiers représentés en binaire sur 1 bit. La représentation de la somme comporte un bit d'unité et un bit de retenue.

a	b	bit de retenue de $a + b$	bit d'unité de $a + b$
0	0		
0	1		
1	0		
1	1		

- b. Compléter la fonction `addition1bit(a, b)` qui retourne le bit de retenue et le bit d'unité de l'addition de deux bits a et b. Tester avec la fonction `test1()`.

```

1 def addition1bit(a, b):
2     unite = .....
3     retenue = .....
4     return retenue, unite
5
6 def test1():
7     """Fonction de test de l'additionneur sur 1 bit"""
8     for a in range(0, 2):
9         for b in range(0, 2):
10            retenue, unite = addition1bit(a, b)
11            print(a, '+', b, '=', str(retenue)+str(unite))

```

2. Dans cette question on fixe $N = 2$ bits.

Quelle est la somme sur deux bits des entiers dont les listes de chiffres en base 2 sont $c = [1, 1]$ et $d = [0, 1]$?

Peut-on utiliser la fonction `addition1bit(a, b)` ci-dessus pour calculer le bit des dizaines de la somme ?

Sinon, comment faudrait-il la modifier ?

Si on ne peut coder que 2 bits de la somme, que peut-on dire du résultat obtenu ?

```

1 In [7]: c = [1,1]
2
3 In [8]: d = [0, 1]
4
5 In [9]: retenue, unite = addition1bit(a[1], b[1])
6
7 In [10]: retenue, unite
8 Out[10]: (1, 0)
9
10 In [11]: int('11', 2) + int('01', 2)
11 Out[11]: 4

```

3. Compléter la fonction `binairer(n)` calculant la liste des chiffres d'un entier n en base 2 sur N bits.

```

1 def binairer(n):
2     """liste des bits (max N) de n"""
3     t = [0]*N #tableau de N bits initialisé à 0 partout
4     k = N - 1

```

4. Compléter la fonction `decimale(n)` calculant la liste des chiffres d'un entier n en base 2 sur N bits.

Un test pour vérifier en console si les deux fonctions sont correctes :

5. Compléter la fonction `addition_listes(L, M)` ci-dessous où `L` et `M` sont deux listes de 0 ou 1 et qui retourne la liste obtenue par somme bit par bit avec retenue, sur `N` bits.

6. Ecrire une fonction `addition` qui calcule l'addition de deux entiers naturels sur N bits. Tester la fonction sur quelques exemples et donner des cas de dépassement de capacité.

```
1
2 In [10]: decimale(addition(4,5))
3 Out[10]: 9
```



Image tirée du site <http://xkcd.com/>