Introduction à la science informatique

Semaine 3

(2 séances de 2 heures)

```python
def somme(n):
    s = 0
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

```python
def somme(n):
    s = 0
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

```python
def somme(n):
    s = 0
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

Comment Python calcule,
en particulier avec des entiers ?

```python
def somme(n):
    s = 0
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

Comment Python calcule,
en particulier avec des entiers ?

Comment les machines
représentent les entiers ?…
pour facilement calculer avec !

Représenter un entier

Représenter un entier

5

23

42

XXIV

vingt-trois

3 198 256

$\mathbf{10^{29}}$

Bien représenter pour calculer facilement

$$\begin{array}{r} 234 \\ +\ 281 \\ \hline 515 \end{array}$$

Bien représenter pour calculer facilement

$$\begin{array}{r} 234 \\ +\ 281 \\ \hline 515 \end{array}$$

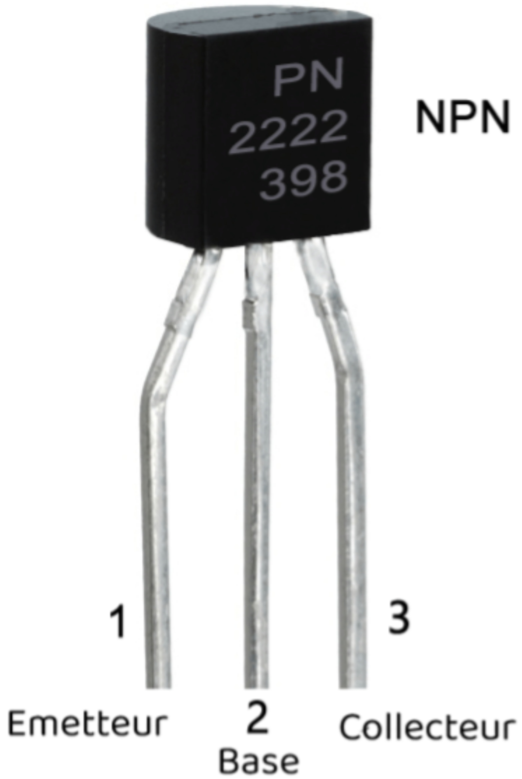
$$\begin{array}{rr} & \text{deux cent trente-quatre} \\ + & \text{deux cent quatre-vingt-un} \\ \hline \end{array}$$

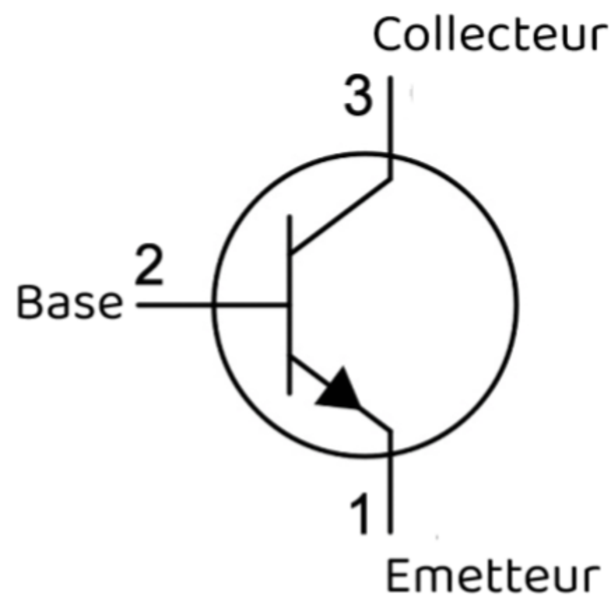
Bien représenter pour calculer facilement

$$\begin{array}{r} 234 \\ +\ 281 \\ \hline 515 \end{array}$$

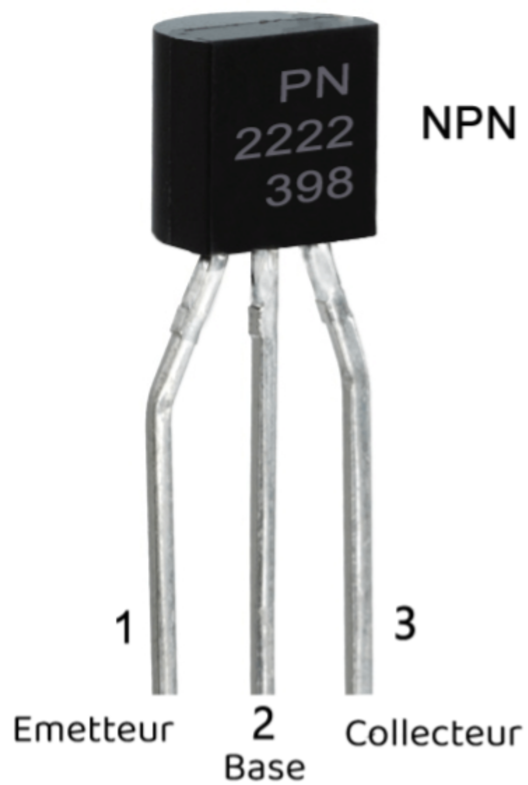
	deux cent trente-quatre
+	deux cent quatre-vingt-un
	n'importe quoi !

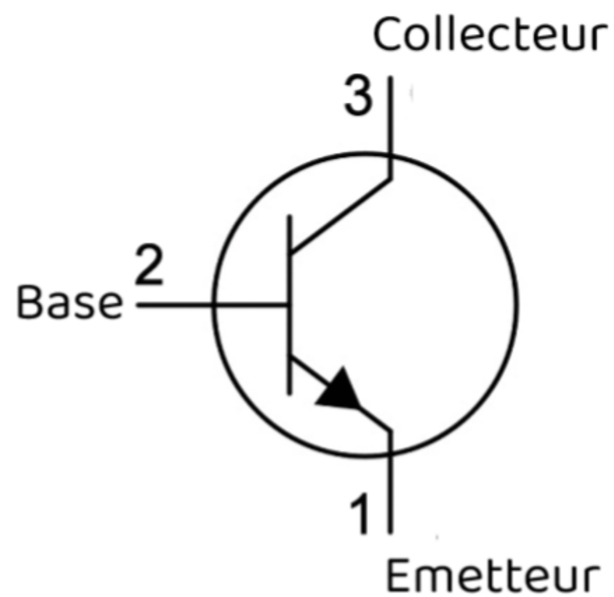
Calculer dans un ordinateur





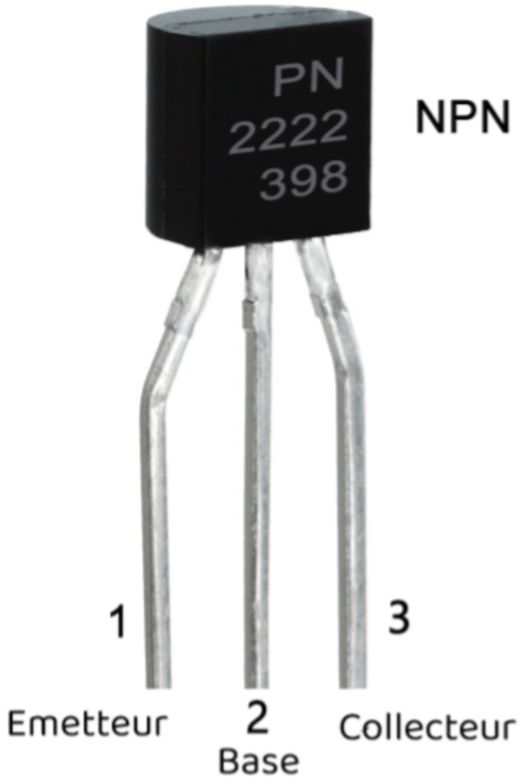
Calculer dans un ordinateur

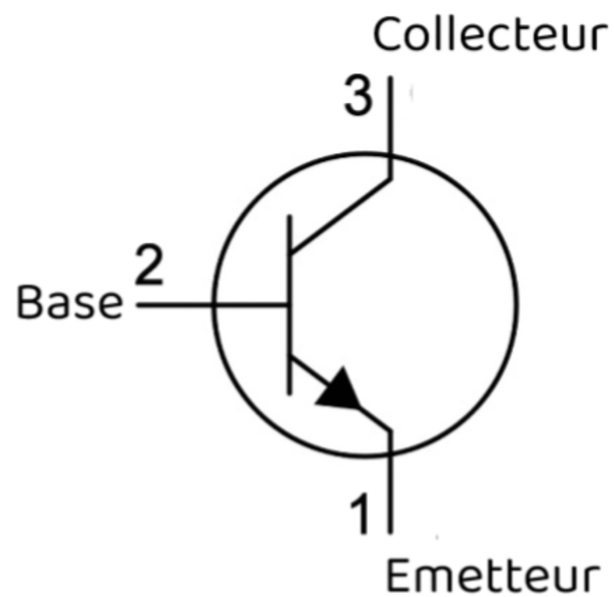




Élément électronique de base de l’ordinateur :
les transistors, qui fonctionnent comme des interrupteurs

Calculer dans un ordinateur





Élément électronique de base de l’ordinateur :
les transistors, qui fonctionnent comme des interrupteurs

Représenter des entiers ?

Représenter des entiers ?

0

Représenter des entiers ?

0
1

Représenter des entiers ?

0
1
2

Représenter des entiers ?

0
1
2
3

Représenter des entiers ?

0

1

2

3

...

Représenter des entiers ?

0
1
2
3
…
9

Représenter des entiers ?

0
1
2
3
...
9
10

Représenter des entiers ?

0
1
2
3
...
9
10
11

Représenter des entiers ?

0
1
2
3
…
9
10
11
…

Représenter des entiers ?

0
1
2
3
…
9
10
11
…
99

Représenter des entiers ?

0
1
2
3
…
9
10
11
…
99
100

Représenter des entiers ?

0
1
2
3
…
9
10
11
…
99
100
101

Représenter des entiers ?

0
1
2
3
…
9
10
11
…
99
100
101

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	
2	
3	
…	
9	
10	
11	
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	
3	
…	
9	
10	
11	
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	
…	
9	
10	
11	
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	
9	
10	
11	
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	
10	
11	
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	
11	
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	110
11	
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	110
11	111
…	
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	110
11	111
…	1000
99	
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	110
11	111
…	1000
99	1001
100	
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	110
11	111
…	1000
99	1001
100	1010
101	

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	110
11	111
…	1000
99	1001
100	1010
101	…

Chiffre des unités,
puis des dizaines,
puis des centaines…

Représenter des entiers ?

0	0
1	1
2	10
3	11
…	100
9	101
10	110
11	111
…	1000
99	1001
100	1010
101	…

Chiffre des unités,
puis des dizaines,
puis des centaines…

Chiffre des unités,
puis de la puissance 2,
puis de la puissance 4…

Notation positionnelle

en décimal…

$$317 = 300 + 10 + 7$$
$$= 3 \times 10^2 + 1 \times 10^1 + 7 \times 10^0$$

Notation positionnelle

en décimal…

$$317 = 300 + 10 + 7$$
$$= 3 \times 10^2 + 1 \times 10^1 + 7 \times 10^0$$

en binaire…

$$100111101 = 1 \times 2^8 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0$$
$$= 256 + 32 + 16 + 8 + 4 + 1$$
$$= 317$$

Notation positionnelle

en décimal…

$$317 = 300 + 10 + 7$$
$$= 3 \times 10^2 + 1 \times 10^1 + 7 \times 10^0$$

Chaque chiffre 0 ou 1 s'appelle un bit

en binaire…

$$100111101 = 1 \times 2^8 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0$$
$$= 256 + 32 + 16 + 8 + 4 + 1$$
$$= 317$$

Notation positionnelle

en décimal…

$$317 = 300 + 10 + 7$$
$$= 3 \times 10^2 + 1 \times 10^1 + 7 \times 10^0$$

Chaque chiffre 0 ou 1 s'appelle un <u>bit</u>

en binaire…

$$100111101 = 1 \times 2^8 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0$$
$$= 256 + 32 + 16 + 8 + 4 + 1$$
$$= 317$$

Pour distinguer la base, on écrit $317_{10} = 100111101_2$

Les puissances de 2

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

Les puissances de 2

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

Les puissances de 2

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

75 = 64 + 11

Les puissances de 2

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

$75 \quad = 64 + 11 \quad = 64 + 8 + 3$

Les puissances de 2

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

75 = 64 + 11 = 64 + 8 + 3 = 64 + 8 + 2 + 1

Les puissances de 2

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

75 = 64 + 11 = 64 + 8 + 3 = 64 + 8 + 2 + 1

donc $75_{10} = 1001011_2$

Trouver l’écriture en base 2 via les divisions euclidiennes

75

Trouver l’écriture en base 2 via les divisions euclidiennes

75	2
1	37

Trouver l’écriture en base 2 via les divisions euclidiennes

75	2	
1	37	2
	1	18

Trouver l’écriture en base 2 via les divisions euclidiennes

$$
\begin{array}{rrrr}
75 & \multicolumn{1}{|l}{2} & & \\ \cline{2-2}
1 & \multicolumn{1}{|l}{37} & \multicolumn{1}{|l}{2} & \\ \cline{3-3}
 & 1 & \multicolumn{1}{|l}{18} & \multicolumn{1}{|l}{2} \\ \cline{4-4}
 & & 0 & \multicolumn{1}{|l}{9}
\end{array}
$$

Trouver l’écriture en base 2 via les divisions euclidiennes

75	2			
1	37	2		
	1	18	2	
		0	9	2
			1	4

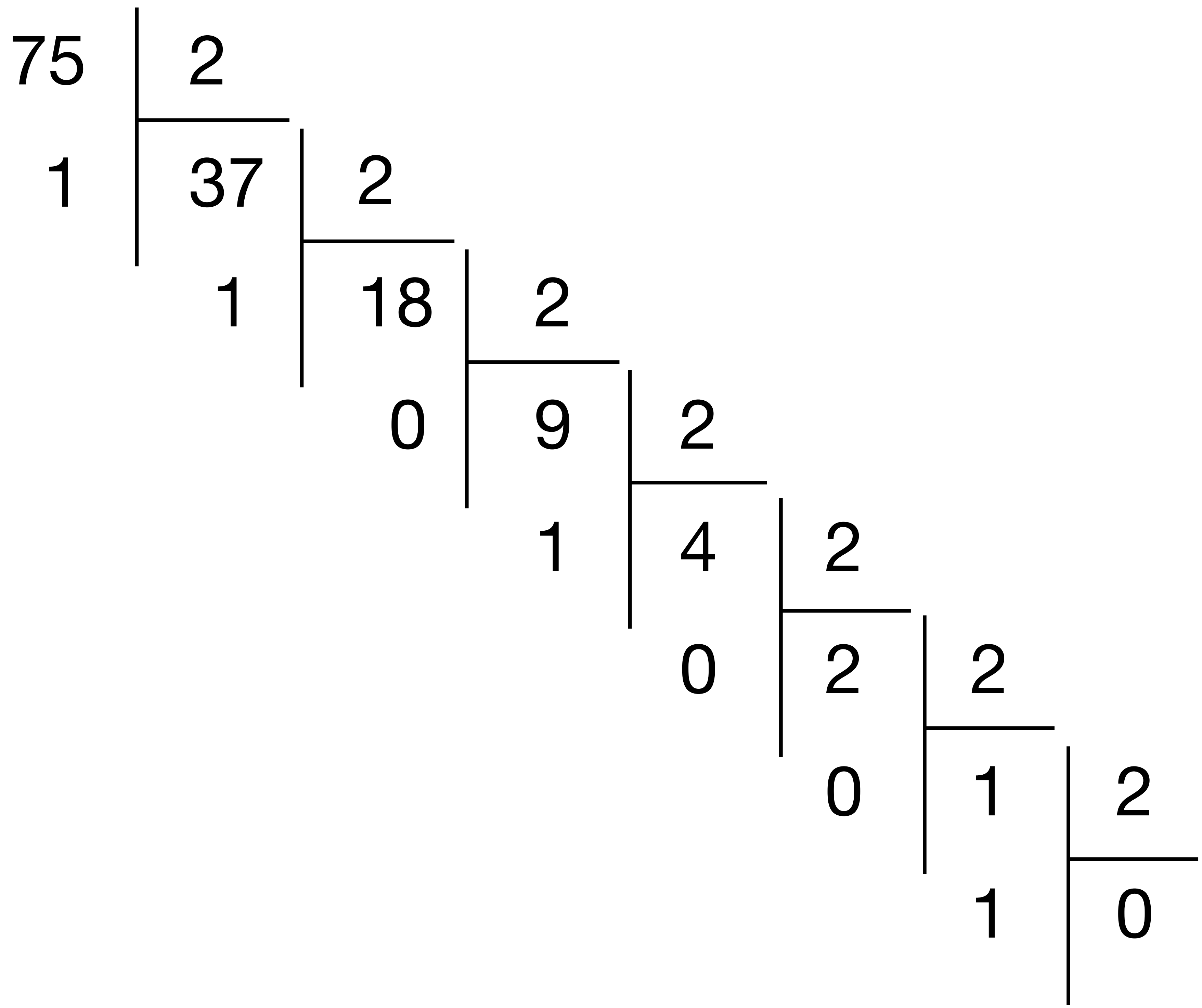
Trouver l’écriture en base 2 via les divisions euclidiennes

75	2				
1	37	2			
	1	18	2		
		0	9	2	
			1	4	2
				0	2

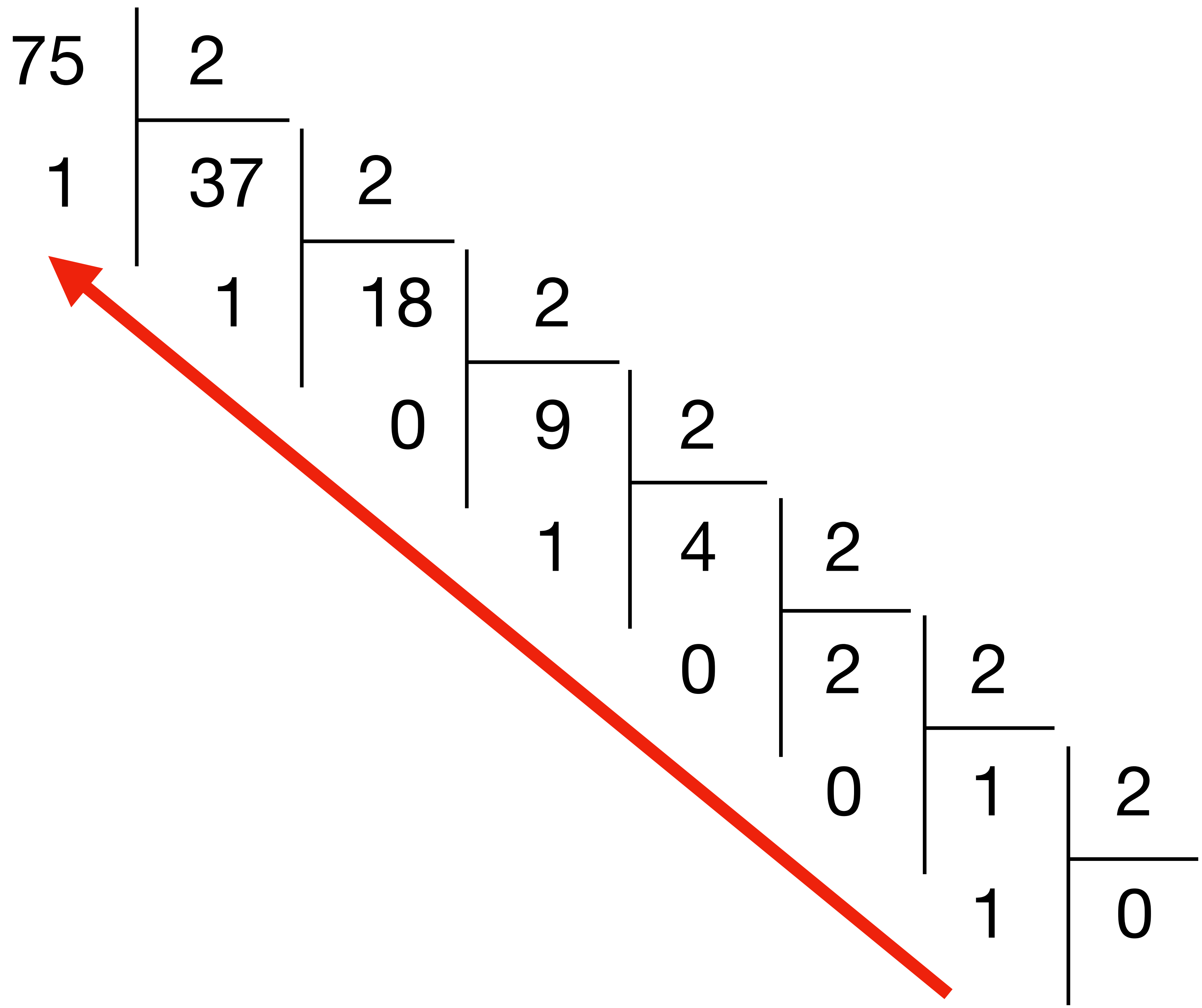
Trouver l’écriture en base 2 via les divisions euclidiennes

75	2					
1	37	2				
	1	18	2			
		0	9	2		
			1	4	2	
				0	2	2
					0	1

Trouver l'écriture en base 2 via les divisions euclidiennes



Trouver l’écriture en base 2 via les divisions euclidiennes



Trouver l’écriture en base 2 via les divisions euclidiennes

75	2	1	37
37	2	1	18
18	2	0	9
9	2	1	4
4	2	0	2
2	2	0	1
1	2	1	0

donc $\quad 75_{10} = 1001011_2$

Exercice 1

Calcul en base 2

$$\begin{array}{r} {\scriptstyle 1} \\ 0\,2\,3 \\ +\,1\,9\,0 \\ \hline 2\,1\,3 \end{array}$$

Calcul en base 2

$$\begin{array}{r} {\scriptstyle 1} \\ 023 \\ +190 \\ \hline 213 \end{array}$$

$$\begin{array}{r} 00010111 \\ +10111110 \\ \hline \end{array}$$

Calcul en base 2

$$\begin{array}{r} {\scriptstyle 1} \\ 0\,2\,3 \\ +\,1\,9\,0 \\ \hline 2\,1\,3 \end{array} \qquad\qquad \begin{array}{r} 0\,0\,0\,1\,0\,1\,1\,1 \\ +\,1\,0\,1\,1\,1\,1\,1\,0 \\ \hline 1 \end{array}$$

Calcul en base 2

$$\begin{array}{cccc} & 1 & & \\ & 0 & 2 & 3 \\ + & 1 & 9 & 0 \\ \hline & 2 & 1 & 3 \end{array}$$

$$\begin{array}{ccccccccc} & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline & & & & & & & 0 & 1 \end{array}$$

Calcul en base 2

$$
\begin{array}{cccc}
 & 1 & & \\
 & 0 & 2 & 3 \\
+ & 1 & 9 & 0 \\
\hline
 & 2 & 1 & 3
\end{array}
\qquad\qquad
\begin{array}{ccccccccc}
 & & & & & & 1 & & \\
 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & & & & & & & 0 & 1
\end{array}
$$

Calcul en base 2

$$\begin{array}{r} {\scriptstyle 1} \\ 023 \\ +190 \\ \hline 213 \end{array} \qquad \begin{array}{r} {\scriptstyle 1} \\ 00010111 \\ +10111110 \\ \hline 101 \end{array}$$

Calcul en base 2

$$
\begin{array}{r}
{\scriptstyle 1} \\
023 \\
+190 \\
\hline
213
\end{array}
\qquad\qquad
\begin{array}{r}
{\scriptstyle 1\,1} \\
00010111 \\
+10111110 \\
\hline
101
\end{array}
$$

Calcul en base 2

$$
\begin{array}{cccc}
 & 1 & & \\
 & 0 & 2 & 3 \\
+ & 1 & 9 & 0 \\
\hline
 & 2 & 1 & 3
\end{array}
\qquad\qquad
\begin{array}{ccccccccc}
 & & & & & 1 & 1 & & \\
 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & & & & & 0 & 1 & 0 & 1
\end{array}
$$

Calcul en base 2

$$
\begin{array}{cccc}
 & {\scriptstyle 1} & & \\
 & 0 & 2 & 3 \\
+ & 1 & 9 & 0 \\
\hline
 & 2 & 1 & 3
\end{array}
\qquad\qquad
\begin{array}{ccccccccc}
 & & & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & \\
 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & & & & & 0 & 1 & 0 & 1
\end{array}
$$

Calcul en base 2

$$\begin{array}{cccc} & 1 & & \\ & 0 & 2 & 3 \\ + & 1 & 9 & 0 \\ \hline & 2 & 1 & 3 \end{array}$$

$$\begin{array}{ccccccccc} & & & & 1 & 1 & 1 & & \\ & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline & & & & 1 & 0 & 1 & 0 & 1 \end{array}$$

Calcul en base 2

$$\begin{array}{cccc} & {\scriptstyle 1} & & \\ & 0 & 2 & 3 \\ + & 1 & 9 & 0 \\ \hline & 2 & 1 & 3 \end{array} \qquad \begin{array}{ccccccccc} & & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & \\ & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline & & & & 1 & 0 & 1 & 0 & 1 \end{array}$$

Calcul en base 2

$$\begin{array}{cccc}
 & 1 & & \\
 & 0 & 2 & 3 \\
+ & 1 & 9 & 0 \\
\hline
 & 2 & 1 & 3
\end{array}
\qquad\qquad
\begin{array}{ccccccccc}
 & & & 1 & 1 & 1 & 1 & & \\
 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & & & 0 & 1 & 0 & 1 & 0 & 1
\end{array}$$

Calcul en base 2

$$\begin{array}{cccc} & 1 & & \\ & 0 & 2 & 3 \\ + & 1 & 9 & 0 \\ \hline & 2 & 1 & 3 \end{array}$$

$$\begin{array}{ccccccccc} & & 1 & 1 & 1 & 1 & 1 & & \\ & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline & & & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

Calcul en base 2

$$\begin{array}{cccc} & 1 & & \\ & 0 & 2 & 3 \\ + & 1 & 9 & 0 \\ \hline & 2 & 1 & 3 \end{array}$$

$$\begin{array}{ccccccccc} & & 1 & 1 & 1 & 1 & 1 & & \\ & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline & & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

Calcul en base 2

$$\begin{array}{cccc}
 & 1 & & \\
 & 0 & 2 & 3 \\
+ & 1 & 9 & 0 \\
\hline
 & 2 & 1 & 3
\end{array}$$

$$\begin{array}{ccccccccc}
 & & 1 & 1 & 1 & 1 & 1 & & \\
 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}$$

Calcul en base 2

$$
\begin{array}{r}
12 \\
\times \quad 11 \\
\hline
12 \\
12 \\
\hline
132
\end{array}
$$

Calcul en base 2

$$
\begin{array}{cccc}
 & & 1 & 2 \\
\times & & 1 & 1 \\
\hline
 & & 1 & 2 \\
 & 1 & 2 & \\
\hline
 & 1 & 3 & 2
\end{array}
\qquad\qquad
\begin{array}{ccccc}
 & 1 & 1 & 0 & 0 \\
\times & 1 & 0 & 1 & 1 \\
\hline
\end{array}
$$

Calcul en base 2

$$
\begin{array}{rrr}
 & 1 & 2 \\
\times & 1 & 1 \\
\hline
 & 1 & 2 \\
1 & 2 & \\
\hline
1 & 3 & 2
\end{array}
\qquad\qquad
\begin{array}{rcccc}
 & 1 & 1 & 0 & 0 \\
\times & 1 & 0 & 1 & 1 \\
\hline
 & 1 & 1 & 0 & 0
\end{array}
$$

Calcul en base 2

$$\begin{array}{r}
12 \\
\times \quad 11 \\
\hline
12 \\
12 \\
\hline
132
\end{array}
\qquad\qquad
\begin{array}{r}
1100 \\
\times \quad 1011 \\
\hline
1100 \\
1100
\end{array}$$

Calcul en base 2

$$
\begin{array}{rrrr}
 & & 1 & 2 \\
\times & & 1 & 1 \\
\hline
 & & 1 & 2 \\
 & 1 & 2 & \\
\hline
 & 1 & 3 & 2
\end{array}
\qquad\qquad
\begin{array}{rrrrrrrr}
 & & & & 1 & 1 & 0 & 0 \\
\times & & & & 1 & 0 & 1 & 1 \\
\hline
 & & & & 1 & 1 & 0 & 0 \\
 & & & 1 & 1 & 0 & 0 & \\
 & & 0 & 0 & 0 & 0 & &
\end{array}
$$

Calcul en base 2

$$
\begin{array}{cccc}
 & & 1 & 2 \\
\times & & 1 & 1 \\
\hline
 & & 1 & 2 \\
 & 1 & 2 & \\
\hline
 & 1 & 3 & 2
\end{array}
\qquad\qquad
\begin{array}{cccccccc}
 & & & & 1 & 1 & 0 & 0 \\
\times & & & & 1 & 0 & 1 & 1 \\
\hline
 & & & & 1 & 1 & 0 & 0 \\
 & & & 1 & 1 & 0 & 0 & \\
 & & 0 & 0 & 0 & 0 & & \\
 & 1 & 1 & 0 & 0 & & &
\end{array}
$$

Calcul en base 2

$$
\begin{array}{r}
12\\
\times\quad 11\\
\hline
12\\
12\\
\hline
132
\end{array}
\qquad\qquad
\begin{array}{r}
1100\\
\times\quad 1011\\
\hline
{}^{1}\,1100\\
{}^{1}\,1100\\
{}^{1}\,0000\\
1100\\
\hline
10000100
\end{array}
$$

Exercices 2 et 3

Complexité de l'addition ?

$$
\begin{array}{cccc}
 & {\scriptstyle 1} & & \\
 & \color{gray}{0} & 2 & 3 \\
+ & 1 & 9 & 0 \\
\hline
 & 2 & 1 & 3
\end{array}
\qquad\qquad
\begin{array}{ccccccccc}
 & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & \\
 & \color{gray}{0} & \color{gray}{0} & \color{gray}{0} & 1 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}
$$

Complexité de l’addition ?

$$\begin{array}{cccc} & 1 & & \\ & 0 & 2 & 3 \\ + & 1 & 9 & 0 \\ \hline & 2 & 1 & 3 \end{array}
\qquad\qquad
\begin{array}{ccccccccc} & & 1 & 1 & 1 & 1 & 1 & & \\ & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

4 opérations élémentaires

Complexité de l'addition ?

$$\begin{array}{cccc}
 & 1 & & \\
 & 0 & 2 & 3 \\
+ & 1 & 9 & 0 \\
\hline
 & 2 & 1 & 3
\end{array}
\qquad\qquad
\begin{array}{ccccccccc}
 & & 1 & 1 & 1 & 1 & 1 & & \\
 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}$$

4 opérations élémentaires

13 opérations élémentaires

Complexité de l'addition ?

$$\begin{array}{r} {\scriptstyle 1} \\ 023 \\ +190 \\ \hline 213 \end{array} \qquad\qquad \begin{array}{r} {\scriptstyle 1\,1\,1\,1\,1} \\ 00010111 \\ +10111110 \\ \hline 11010101 \end{array}$$

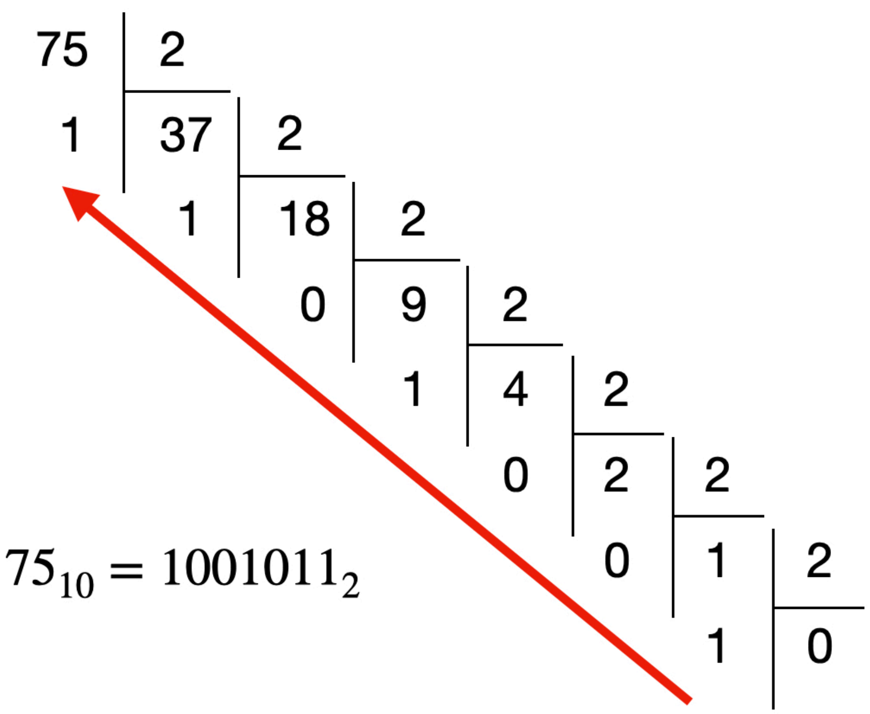
4 opérations élémentaires

13 opérations élémentaires

Dans le pire des cas, 2 fois le nombre maximal de chiffres + 1

(cf TD 1)

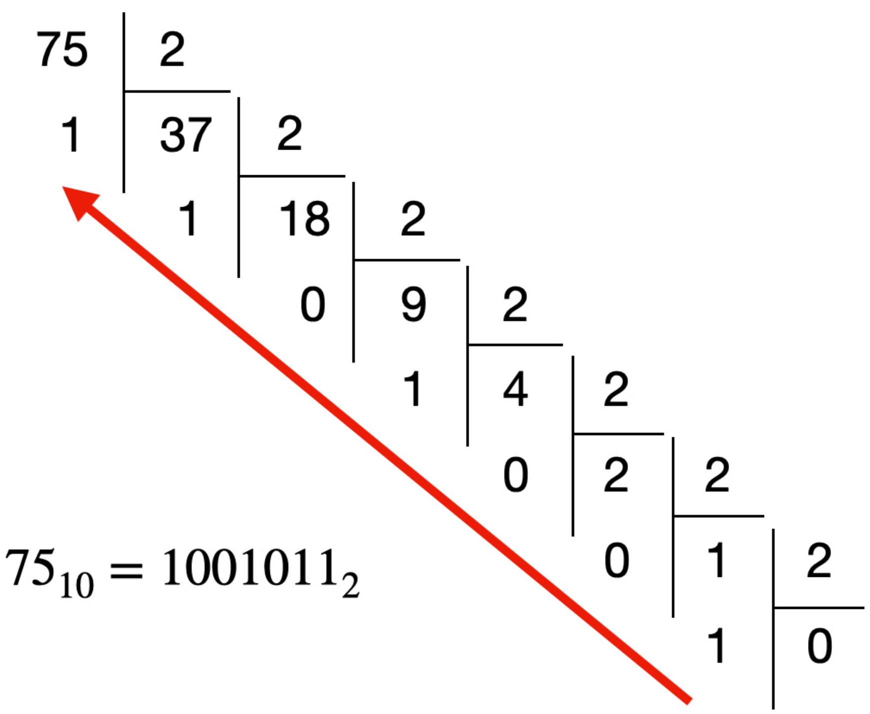
Nombre de bits ?



donc $75_{10} = 1001011_2$

« Le nombre de bits de la représentation binaire de l'entier *n* est le nombre de fois qu'on peut le diviser par 2 avant d'arriver à 0. »

Nombre de bits ?



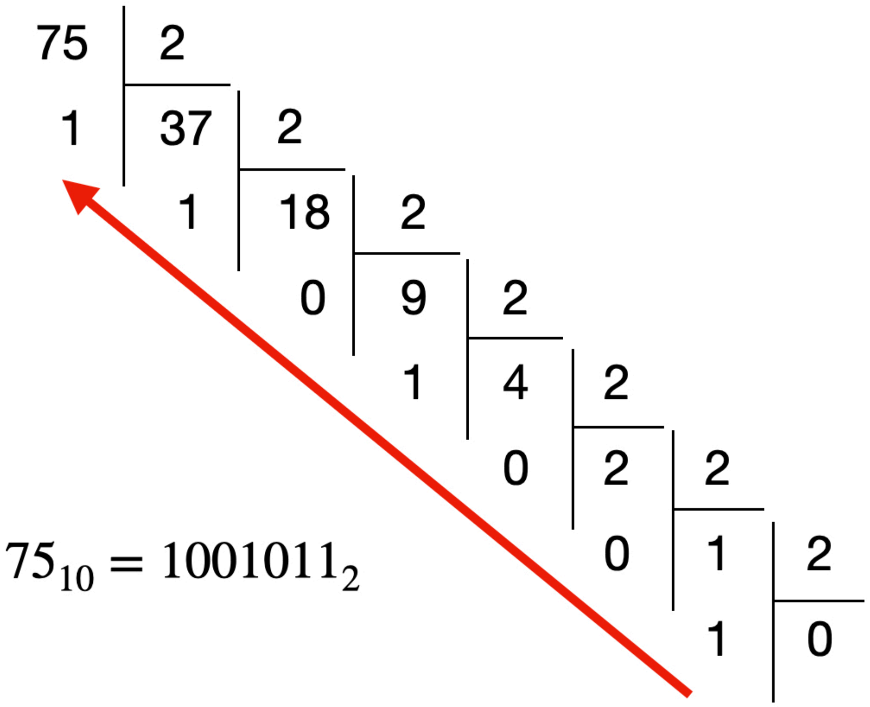
donc $75_{10} = 1001011_2$

« Le nombre de bits de la représentation binaire de l'entier *n* est le nombre de fois qu'on peut le diviser par 2 avant d'arriver à 0. »

Sur *k* bits, on peut compter jusqu'à

$$1 + 2^1 + 2^2 + \cdots + 2^{k-1} = 2^k - 1$$

Nombre de bits ?



donc $75_{10} = 1001011_2$

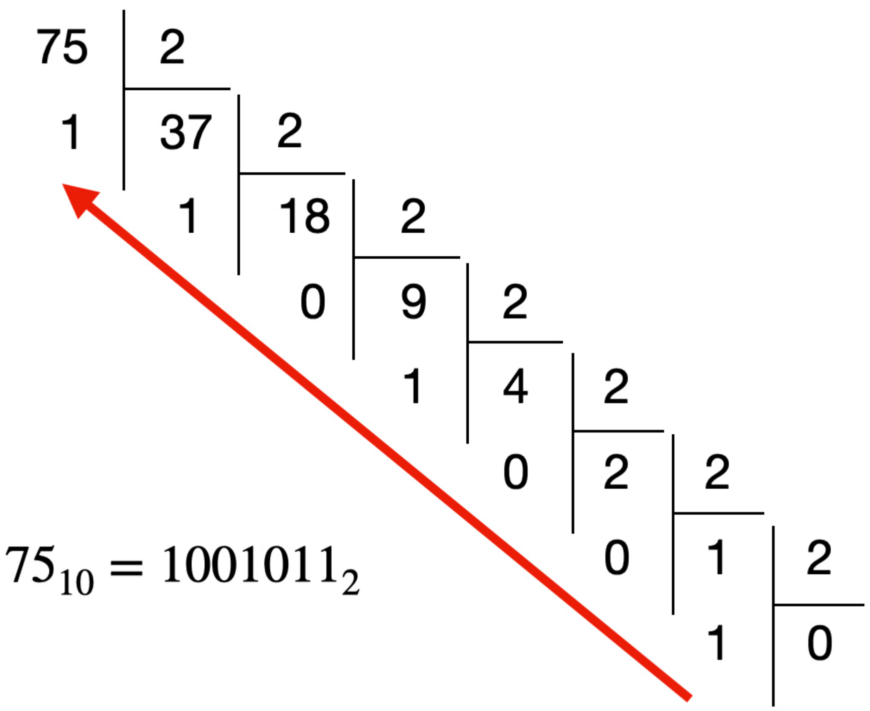
« Le nombre de bits de la représentation binaire de l'entier *n* est le nombre de fois qu'on peut le diviser par 2 avant d'arriver à 0. »

Sur *k* bits, on peut compter jusqu'à

$$1 + 2^1 + 2^2 + \cdots + 2^{k-1} = 2^k - 1$$

Donc si $2^{k-1} \leq n < 2^k$, la représentation binaire de l'entier *n* contient *k* bits. Ceci équivaut à $k \leq \log_2(n) + 1 < k + 1$.

Nombre de bits ?



donc $75_{10} = 1001011_2$

« Le nombre de bits de la représentation binaire de l'entier *n* est le nombre de fois qu'on peut le diviser par 2 avant d'arriver à 0. »

Sur *k* bits, on peut compter jusqu'à

$$1 + 2^1 + 2^2 + \cdots + 2^{k-1} = 2^k - 1$$

Donc si $2^{k-1} \leq n < 2^k$, la représentation binaire de l'entier *n* contient *k* bits. Ceci équivaut à $k \leq \log_2(n) + 1 < k + 1$.

Théorème : Si *n* est un entier strictement positif, sa représentation binaire contient $\lfloor \log_2(n) + 1 \rfloor$ bits.

Dénombrement…

1. Comptons-nous !

2. Compter deux par deux…

3. Compter cinq par cinq…

4. Distribuons le calcul…

Entrée *: des étudiants debout dans un amphi*

- Chaque étudiant a en tête le nombre 1
*- **Tant qu**'il reste au moins deux étudiants debout :*

- chaque étudiant encore debout cherche du regard un autre étudiant debout
- les deux étudiants s'échangent le nombre qu'ils ont en tête (indépendamment des autres étudiants)
- l'un des deux étudiants s'assoit ; l'autre étudiant additionne les deux nombres et reste debout

Sortie *: le dernier étudiant debout crie son nombre*

4. Distribuons le calcul…

Entrée *: des étudiants debout dans un amphi*

- Chaque étudiant a en tête le nombre 1
*- **Tant qu**'il reste au moins deux étudiants debout :*
- chaque étudiant encore debout cherche du regard un autre étudiant debout
- les deux étudiants s'échangent le nombre qu'ils ont en tête (indépendamment des autres étudiants)
- l'un des deux étudiants s'assoit ; l'autre étudiant additionne les deux nombres et reste debout

Sortie *: le dernier étudiant debout crie son nombre*

Nombre d'étapes si *n* étudiants : environ le nombre de fois qu'on peut diviser *n* par 2 avant d'arriver à 1

donc $\lfloor \log_2(n) + 1 \rfloor - 1 = \lfloor \log_2(n) \rfloor$ étapes

1. Comptons-nous !

2. Compter deux par deux…

3. Compter cinq par cinq…

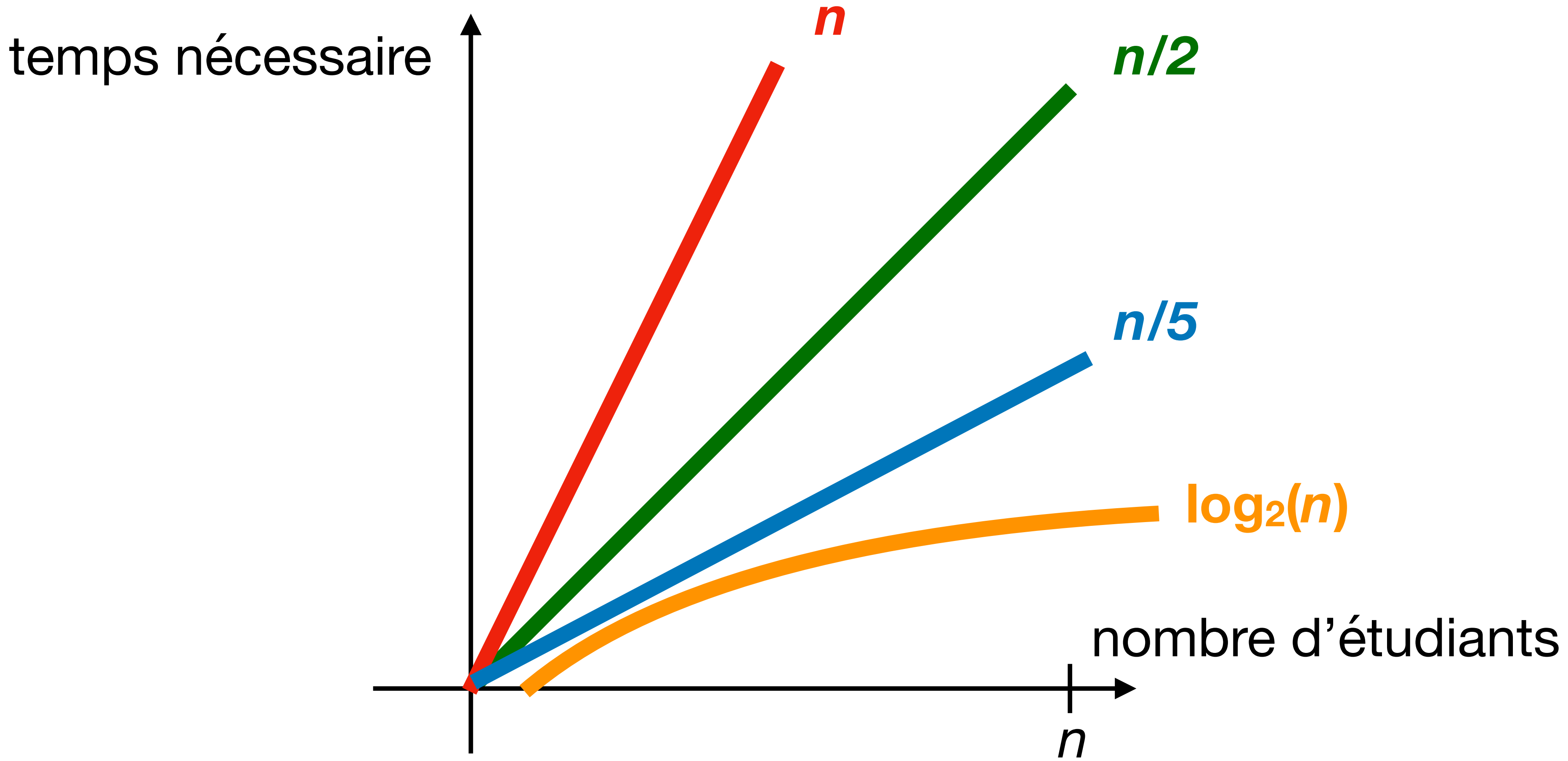
4. Distribuons le calcul…

1. Comptons-nous !

2. Compter deux par deux…

3. Compter cinq par cinq…

4. Distribuons le calcul…



Exercice 4

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):
    s = 0
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):                    1 op
    s = 0
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):          1 op
    s = 0              1 op
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):               1 op
    s = 0                   1 op
    i = 1                   1 op
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):          1 op
    s = 0              1 op
    i = 1              1 op
    while i <= n:      1 op
        s = s + i
        i = i + 1
    return s
```

Complexité ?

On compte le nombre d’opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):          1 op
    s = 0              1 op
    i = 1              1 op
    while i <= n:      1 op
        s = s + i      1 op
        i = i + 1
    return s
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):           1 op
    s = 0               1 op
    i = 1               1 op
    while i <= n:       1 op
        s = s + i       1 op
        i = i + 1       1 op
    return s
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):          1 op
    s = 0              1 op
    i = 1              1 op
    while i <= n:      1 op
        s = s + i      1 op
        i = i + 1      1 op
    return s           1 op
```

Complexité ?

On compte le nombre d’opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):         1 op   × 1 fois
    s = 0             1 op
    i = 1             1 op
    while i <= n:     1 op
        s = s + i     1 op
        i = i + 1     1 op
    return s          1 op
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```
def somme(n):           1 op   × 1 fois
    s = 0               1 op   × 1 fois
    i = 1               1 op
    while i <= n:       1 op
        s = s + i       1 op
        i = i + 1       1 op
    return s            1 op
```

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

`def somme(n):`	1 op	× 1 fois
`    s = 0`	1 op	× 1 fois
`    i = 1`	1 op	× 1 fois
`    while i <= n:`	1 op	
`        s = s + i`	1 op	
`        i = i + 1`	1 op	
`    return s`	1 op	

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

`def somme(n):`	1 op	$\times$ 1 fois
`    s = 0`	1 op	$\times$ 1 fois
`    i = 1`	1 op	$\times$ 1 fois
`    while i <= n:`	1 op	
`        s = s + i`	1 op	
`        i = i + 1`	1 op	
`    return s`	1 op	$\times$ 1 fois

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

`def somme(n):`	1 op	$\times$ 1 fois
`    s = 0`	1 op	$\times$ 1 fois
`    i = 1`	1 op	$\times$ 1 fois
`    while i <= n:`	1 op	
`        s = s + i`	1 op	$\times\, n$ fois
`        i = i + 1`	1 op	
`    return s`	1 op	$\times$ 1 fois

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

`def somme(n):`	1 op	$\times$ 1 fois
`    s = 0`	1 op	$\times$ 1 fois
`    i = 1`	1 op	$\times$ 1 fois
`    while i <= n:`	1 op	
`        s = s + i`	1 op	$\times\, n$ fois
`        i = i + 1`	1 op	$\times\, n$ fois
`    return s`	1 op	$\times$ 1 fois

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

```python
def somme(n):
    s = 0
    i = 1
    while i <= n:
        s = s + i
        i = i + 1
    return s
```

`def somme(n):`	1 op	$\times$ 1 fois
`s = 0`	1 op	$\times$ 1 fois
`i = 1`	1 op	$\times$ 1 fois
`while i <= n:`	1 op	$\times (n+1)$ fois
`s = s + i`	1 op	$\times n$ fois
`i = i + 1`	1 op	$\times n$ fois
`return s`	1 op	$\times$ 1 fois

Complexité ?

On compte le nombre d'opérations par ligne (1 sauf si appel de fonction) et on compte combien de fois cette ligne est exécutée…

`def somme(n):`	1 op	$\times$ 1 fois
`    s = 0`	1 op	$\times$ 1 fois
`    i = 1`	1 op	$\times$ 1 fois
`    while i <= n:`	1 op	$\times\ (n+1)$ fois
`        s = s + i`	1 op	$\times\ n$ fois
`        i = i + 1`	1 op	$\times\ n$ fois
`    return s`	1 op	$\times$ 1 fois

$$= 3n + 5$$ opérations

D’autres représentations ?

D’autres représentations ?

- Décimal : base 10
- Binaire : base 2

D’autres représentations ?

- Décimal : base 10
- Binaire : base 2
- Hexadécimal : base 16

Base 16 : hexadécimal

n	0	1	2	3	4
16^n	1	16	256	4096	65536

$$317 = 256 + 3 \times 16 + 13$$

$$= 1 \times 16^2 + 3 \times 16^1 + 13 \times 16^0$$

Chiffres en base 16 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *A, B, C, D, E, F*

13 devient un « chiffre » dans l'écriture en base 16, de même que 10, 11, 12, 14 et 15…

Base 16 : hexadécimal

n	0	1	2	3	4
16^n	1	16	256	4096	65536

$$317 = 256 + 3 \times 16 + 13$$

$$= 1 \times 16^2 + 3 \times 16^1 + 13 \times 16^0$$

13 devient un « chiffre » dans l'écriture en base 16, de même que 10, 11, 12, 14 et 15…

Chiffres en base 16 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *A, B, C, D, E, F*

$$317_{10} = 13D_{16}$$

De la base 10 à la base 16

317

De la base 10 à la base 16

317	16
13	19

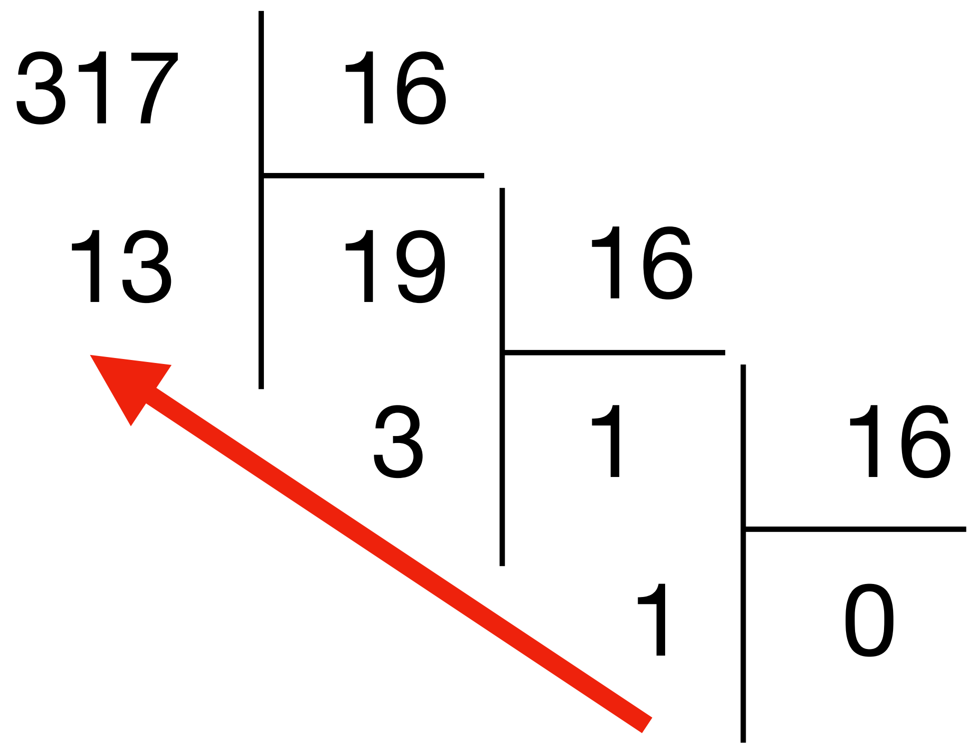
De la base 10 à la base 16

317	16	
13	19	16
	3	1

De la base 10 à la base 16

317	16		
13	19	16	
	3	1	16
		1	0

De la base 10 à la base 16



De la base 10 à la base 16

$$\begin{array}{r|l} 317 & 16 \\ \cline{2-2} 13 & \begin{array}{r|l} 19 & 16 \\ \cline{2-2} 3 & \begin{array}{r|l} 1 & 16 \\ \cline{2-2} 1 & 0 \end{array} \end{array} \end{array}$$

donc $317_{10} = 13D_{16}$

Lien base 16 - base 2

101011100101

AE5

Lien base 16 - base 2

101011100101 AE5

1010 1110 0101

Lien base 16 - base 2

101011100101 AE5

1010 1110 0101

A E 5

Lien base 16 - base 2

101011100101 AE5

1010 1110 0101

A E 5

Grouper les bits par 4 (car $2^4 = 16$) et remplacer par le chiffre en base 16

Lien base 16 - base 2

101011100101 AE5

1010	1110	0101
A	E	5

Grouper les bits par 4 (car $2^4 = 16$) et remplacer par le chiffre en base 16

La base 16 est donc une façon de rendre lisible par des humains du code binaire, par exemple dans les adresses IPv6 (qui contiennent $8 \times 16 = 128$ bits)

2001:0DB8:3C4D:0015:0000:0000:1A2F:1A2B

0010000000000001 0000110110111000 0011110001001101 0000000000010101
0000000000000000 0000000000000000 0001101000101111 0001101000101011

Exercice 5