

---

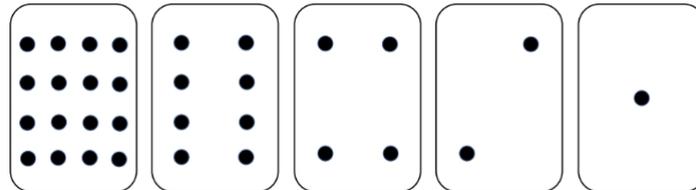
## DES 0 ET DES 1

---

**Objectifs :**

- Savoir écrire un entier naturel en binaire.
- Savoir utiliser l'algorithme de changement de base.

Dans un ordinateur, toutes les informations sont représentées à l'aide de deux chiffres 0 et 1, appelés chiffres binaires ou *Binary Digits* en anglais (ou plus simplement *bits*). Nous allons essayer de comprendre dans un premier temps, comment un ordinateur peut représenter n'importe quel nombre entier positif avec uniquement ces deux chiffres.



**Jouons un peu** Dans cette partie, tu vas te servir d'un jeu de carte un peu particulier pour apprendre à décomposer les entiers, comme tu l'as appris à l'école primaire. Tu vas voir, cela va te rappeler de « bons vieux souvenirs » !

1. Observe bien le jeu de carte. Que remarques-tu à propos du nombre de points qui figure sur les cartes ?
2. On va se servir du jeu de cartes pour représenter des nombres. Voici le principe : retourne certaines cartes face-cachées puis additionne tous les points visibles pour obtenir le nombre souhaité.
  - (a) Représente ainsi les nombres 6, 15 puis 21.
  - (b) Pour pouvoir écrire plus facilement ces nombres, on va adopter la convention d'écriture suivante : en ordonnant les cartes selon un nombre de points croissant de la droite vers la gauche, écris le chiffre 1 si la carte est face visible, 0 si la carte est face-cachée.  
Écris les nombres précédents avec cette convention d'écriture.
3. Comment écrire les nombres 3, 12 et 19 ? Existe-t-il plusieurs moyen pour obtenir un même nombre ?
4. Quel est le plus grand nombre de l'on peut obtenir avec ces cartes ? Et le plus petit ?
5. Y-a-t-il un entier que tu ne puisses pas obtenir entre ces deux extremums ?
6. Trouve quel nombre est représenté avec l'écriture 10111 ? Et avec 11101 ?

**Les algorithmes à la rescousse** La méthode du jeu de carte nous aide bien pour représenter les entiers sous forme de 0 et de 1, tant que le nombre reste relativement petit. Cependant dans le cas d'un grand nombre, comme le nombre de 76 300 000 000 de transistors qu'embarque le processeur monobloc grand public le plus dense du marché, nous serions vite à court de carte... Nous allons donc faire appel aux algorithmes pour nous aider !

```

1 def binaire(n):
2     q = n
3     r = -1
4     while q > 0 :
5         r = q % 2
6         q = q // 2
7         print(r, end="")
8     print()

```

1. Écris la trace d'exécution que produirait l'appel *binaire(29)* puis compare les sorties obtenues avec la représentation de 29 contenue précédemment.
2. Écris ce programme sous la forme d'un algorithme. On utilisera pour cela le langage naturel ; on symbolise l'instruction d'affectation par la flèche  $\leftarrow$ , et on formule les instructions dans le langage courant (le français dans notre cas) en utilisant des verbes d'action.
3. Exécute cet algorithme avec comme donnée d'entrée le nombre 1022 en posant toutes les divisions entières. « à la main ».
4. Sers-toi de la fonction Python pour obtenir la représentation du nombre 76 300 000 000 avec des 1 et des 0. Quel lien peux-tu établir entre le nombre de chiffres obtenus et le nombre de répétition effectuées ?