# Devoir maison N°1

## 1. Un peu de conversion de base

**Q1. Donner** la représentation en base 2 et sur 8 bits des entiers 14, 218, 42 et 57. **Vérifier** vos réponses sur une console Python avec la fonction **`bin`**.

| Valeur décimale | Valeur binaire |
|---|---|
| 14 | 0000 1110 |
| 218 | 1101 1010 |
| 42 | 0010 1010 |
| 57 | 0011 1001 |

*Remarque : Utilisation de la fonction bin*

```
>>> bin(42)  # Renvoie la représentation en binaire de 42
```

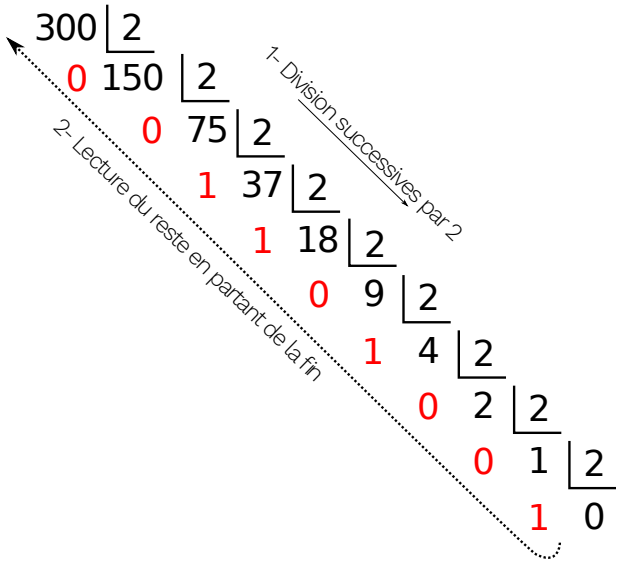
**Q2. Ecrire** en binaire l'entier $300_{10}$ . en détaillant les étapes de calcul. **Convertir** aussi en base 2 les nombres $42_{10}$ et $137_{10}$ . Dans chaque cas, **indiquer** le nombre d'octets nécessaires au codage.

Conversion de 300

méthode 1 : tableau de poids

| 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |

méthode 2 : divisions successives par 2




$300_{(10)} = 0000\ 0001\ 0010\ 1100_{(2)}$ → 2 octets
$42_{(10)} = 0010\ 1010_{(2)}$ → 1 octet
$137_{(10)} = 1000\ 1001_{2)}$ → 1 octet

**Q3. Donner** la représentation en base 16 des entiers binaires suivants. **Vérifier** vos réponses avec la fonction **hex**.

- ➔ $0100\ 1010_{(2)}$ → $4A
- ➔ $1\ 0001\ 0001_{(2)}$ → $111
- ➔ $1010\ 0100\ 1111\ 0010_{(2)}$ → $A4F2

**Q4. Déterminer par calcul** la valeur en base 10 de l'entier qui s'écrit **BEEF** en base 16

$\$BEEF = 11x16^3 + 14x16^2 + 14x16^1 + 15x16^0 = 48879_{(10)}$

## 2.Devine le nombre

Dans le jeu Devine le nombre, un joueur doit trouver le nombre mystère choisi au préalable par le maître du jeu, qui sera ici l'ordinateur. A chaque tour, le joueur propose un nombre et le maître de jeu, indique si le nombre mystère est plus petit ou plus grand que le nombre proposé.
Si le nombre mystère est 5 par exemple, la partie suivante pourra se dérouler :

```
Devine le nombre !
8
plus petit
4
plus grand
5
C'est gagné !
```

Dans ce jeu, l'ordinateur doit tirer au hasard un nombre compris entre deux bornes [a, b]. La fonction **randint(a, b)** de la bibliothèque **random** permet de faire se tirage aléatoire. La syntaxe est la suivante :

```
from random import randint
nbre_aleatoire = randint(10, 20) #nbre_aleatoire : valeur comprise entre 10 et 20
```

**Q5. Ecrire** une fonction **devine_nombre()** permettant de jouer à devine nombre, avec un nombre à chercher compris entre 0 et 100.

## 3.Le jeu des allumettes

Le jeu des allumettes est un jeu de deux joueurs qui se joue ainsi. Initialement, 21 allumettes sont posées sur la table. A tour de rôle, chaque joueur enlève une, deux ou trois allumettes. Celui qui enlève la dernière allumette a perdu.

**Objectif de l'exercice :**

Rechercher un programme afin de jouer au jeu des allumettes contre l'ordinateur.

### Version simpliste du jeu

Une programmation simpliste consiste à définir une boucle **`while`** et d'une variable allumettes contenant le nombre courant d'allumettes. Dans cette boucle l'utilisateur indique combien d'allumettes il enlève puis l'ordinateur enlève un nombre aléatoire d'allumettes (au moins une et pas plus de trois), tant qu'il reste des allumettes.
Si l'on affiche au début de chaque tour de boucle, le nombre actuel d'allumettes, le programme affichera par exemple la séquence suivante :

```
Il y a 21 allumettes
Combien en prenez vous ? 3
je prends 2 allumettes
il y a 16 allumettes
combien en prenez vous ? 1
...
```

**Q6. Ecrire** une fonction **`jeu_allumettes()`** qui respecte ce fonctionnement. Attention à ne pas faire tirer par l'ordinateur plus d'allumettes qu'il n'en reste.
**Q7. Améliorer** le programme pour qu'il affiche **`vous avez perdu`** ou **`vous avez gagné`** lorsque la dernière allumette est retirée, juste avant la sortie de boucle.

# Pour aller plus loin

**Q8. Améliorer** encore ce programme pour qu'il s'assure que le nombre d'allumettes choisi par le joueur est bien au moins égal à un, au plus égal à trois et pas plus grand que le nombre d'allumettes restantes. On pourra pour cela utiliser une boucle **`while`** pour répéter la saisie tant que la valeur n'est pas correcte.

Il se trouve qu'il existe une stratégie gagnante pour le joueur qui joue en second.

**Q9. Trouver** et **programmer** cette stratégie gagnante de manière à ce que l'ordinateur gagne systématiquement.