

## Sujet 10

### □ Exercice 1 : Mot équilibré

On dit qu'un mot est *équilibré* lorsqu'il contient autant de voyelles que de consonnes. Par exemple le mot **rapide** est un mot équilibré, par contre **lent** n'est pas équilibré. Ecrire une fonction `est_equilibre` qui prend en paramètre un mot `mot` et renvoie `True` si `mot` est équilibré et `False` sinon. On suppose que `mot` est constitué uniquement de lettres minuscules et non accentuées et on rappelle que les voyelles sont les lettres **a, e, i, o, u** et **y**.

**Exemples :**

```
1 >>> est_equilibre("oiseau")
2 False
3 >>> est_equilibre("patate")
4 True
5 >>> est_equilibre("toux")
6 True
```

### □ Exercice 2 : Conversion en binaire

Pour rappel, la conversion d'un nombre entier en binaire peut s'effectuer à l'aide de *l'algorithme des divisions successives* qui consiste à :

- effectuer la division euclidienne de  $n$  par 2, soit  $q$  le quotient et  $r$  le reste.
- si le quotient est 0, on s'arrête l'écriture binaire est la suite des restes prises dans l'ordre inverse. Sinon on recommence l'étape 1 en remplaçant  $n$  par  $q$

Par exemple pour  $n = 77$  :

- $77 = 2 \times 38 + 1$ , comme  $38 \neq 0$  on continue
- $38 = 2 \times 19 + 0$ , comme  $19 \neq 0$  on continue
- $19 = 2 \times 9 + 1$
- $9 = 2 \times 4 + 1$
- $4 = 2 \times 2 + 0$
- $2 = 2 \times 1 + 0$
- $1 = 2 \times 0 + 1$  arrêt car le quotient est 0.

L'écriture binaire de 77 est la suite des restes prises dans l'ordre inverse : 1001101.

Compléter la fonction ci-dessous qui implémente cet algorithme, on rappelle que le quotient dans la division euclidienne de  $n$  par 2 s'obtient avec  $n//2$  et le reste avec  $n\%2$  :

```
1 def binaire(n):
2     # On effectue la première division euclidienne, le reste doit être
3     # converti en chaîne de caractères pour initialiser l'écriture binaire
4     ecriture_binaire = str(...)
5     # Tant que le quotient n'est pas nul, on recommence
6     while ..... :
7         ecriture_binaire = str(...) + ....
8         n = .....
9     return ecriture_binaire
```

**Exemples :**

```
1 >>> binaire(77)
2 '1001101'
```