But

Le but est d'écrire soi-même des routines `python` permettant de

- ◇ convertir un nombre entier en son écriture binaire (ou en une autre base) et réciproquement,
- ◇ utiliser ces méthodes pour décoder un texte
- ◇ convertir un nombre non nul quelconque en son écriture flottante 32bits et réciproquement,

Aide

- ◇ Télécharger, dans votre répertoire élève, le fichier `NSI-projet04.py` sur la page [http://forhan.maths.free.fr].
  Le fichier `NSI-projet04.py` contient
  - la phrase `PHCODEE` à décoder
  - la structure de la fonction `entier2binaire(n : int) -> str`
  - la structure de la fonction `flottant2float(stf : str) -> float`
  - la structure de la fonction `entier2flottant(n : int) -> str`
- ◇ Le code `python` ci-contre permet d'obtenir le quotient (q) et le reste (r) de la division euclidienne du nombre n par 2 ; $n = 2 \times q + r$

```
n//2  # renvoie le quotient q de n/2
n%2   # renvoie le reste r de n/2
```

Étapes

a) Adapter le nom du fichier `NSI-projet04.py` au format `NSI-projet04-Nom1Nom2.py`.

b) Écrire une procédure `main() -> None`
qui comportera les tests faits pour tester vos fonctions de conversion : nombre <-> chaîne de caractères.

c) Écrire une fonction `binaire2entier(binaire : str) -> int` qui renvoie l'entier codé en binaire.
e.g. `binaire2entier('1010')` renverra l'entier 10.

d) Déterminer l'écriture binaire du nombre 164 à l'aide d'une succession de division par 2.

Écrire une fonction `entier2binaire(n : int) -> str` qui renvoie le code binaire de l'entier n.
e.g. `entier2binaire(10)` renverra '1010'.

e) Proposer une fonction `decodage1(lettres : str) -> str` qui à partir d'une chaîne de 3 caractères comportant le code ascii renvoie le caractère correspondant.
e.g. `decodage1('077')` renverra le caractère 'M'.

f) Proposer une fonction `decodage2(lettres : str) -> str` qui à partir d'une chaîne de 8 caractères comportant le code binaire du code ascii renvoie le caractère correspondant.
e.g. puisque $\overline{01001101}^2 = 77$, `decodage2('01001101')` renverra aussi le caractère 'M'.
Tester votre fonction (penser à garder ces tests pour le contre-rendu) et appeler votre professeur.

g) Proposer une fonction `decodageG(ph: str) -> None` qui décode
Appelez le professeur à chaque étape du décodage (nb : il y en a trois).

h) Écrire une fonction `flottant2float(stf : str) -> float` qui à partir d'un flottant (32b) sous forme d'une chaîne de caractères renvoie le nombre flottant.
e.g. `floattant2float("0'10000001'01001000000000000000000")` renverra le flottant 5.125.

i) Proposer une fonction `puissance2entier(n : int) -> int` qui renvoie la plus grande puissance de 2 contenu dans n.
e.g. puisque $21 = 2^4 + 2^2 + 2^0$, `puissance2entier(21)` renverra 4.

j) Déterminer l'écriture flottante (32b) de l'entier 21.

Écrire une fonction `entier2flottant(n : int) -> str` qui renvoie la chaîne de caractères codant l'entier n sous forme d'un flottant 32b.

e.g. `entier2flottant(15)` renverra "1'10000010'11100000000000000000000".

k) Approfondissement :

- ◇ adapter les fonctions `entier2binaire()` et `binaire2entier()` en `entier2chaine(n : int, base : int) -> str` et `chaine2entier(ch : str, base : int) -> int` afin de pouvoir traduire un nombre dans n'importe quelle base supérieure à 2.
- ◇ adapter la fonction `puissance2entier()` afin qu'elle fonctionne pour tout flottant positif.
- ◇ adapter la fonction `entier2flottant()` en une fonction `float2flottant()`
- ◇ proposer des fonctions de codage inverses des fonctions de décodage (on fera attention qu'il faut renvoyer un nombre binaire sur 8 bits).
- ◇ proposer une fonction `codageG(ph : str) -> str` qui encode la chaîne de caractères ph selon les fonctions de codage précédemment codées.

Rendu

À la fin du temps imparti, chaque groupe rendra 2 travaux :

a) un compte-rendu (au maximum 2 pages) contenant l'organigramme d'une fonction de conversion et d'une fonction de décodage (au choix), un résumé de votre projet, les jeux de tests effectués ainsi que leurs résultats (ceux-ci pourront être mis dans votre fichier .py)

b) le fichier contenant le code `python` sera envoyé sous format numérique via l'ENT