

## 1. Programme convertisseur de base

Le programme doit réaliser les conversions entre les bases décimal – hexadécimal – binaire.  
Chaque conversion doit correspondre à une fonction.

**Aide** : en fonction des choix effectués pour construire les fonctions de conversion, les fonctions suivantes peuvent être utiles :

- la fonction `str(données)` permet de convertir des données en chaîne de caractères ;
- la fonction `list(chaîne)` permet de convertir une chaîne en liste.

### 1.1. Fonction binaire\_decimal

Elle doit convertir un mot binaire (chaîne de caractères) en un nombre décimal (entier).

```
nombre_decimal = 0
pour indice allant de 0 à la longueur du mot_binaire faire
    si mot_binaire[indice] = 1
        alors nombre_decimal = nombre_decimal + 2(rang du bit)
    fin si
fin pour
```

### 1.2. Fonction decimal\_binaire

Elle doit convertir un nombre décimal (entier) en un mot binaire (chaîne de caractères).

```
mot = vide
mot_binaire = vide
tant que quotient ≠ 0 faire
    reste = reste de quotient / 2
    mot = mot + reste
    quotient = partie entière de quotient / 2
fin tant que
pour indice allant de longueur de mot à 0 faire
    mot_binaire = mot_binaire + mot[indice]
fin pour
```

### 1.3. Fonction hexa\_binaire

Elle convertit un nombre hexadécimal (chaîne de caractères) en mot binaire (chaîne de caractères).

```
tab_hexa = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E", "F"]
tab_binaire = ["0000", "0001", "0010", "0011", "0100", "0101", "0110", "0111", "1000", "1001", "1010", "1011", "1100", "1101", "1110", "1111"]

mot_binaire = vide
pour chiffre dans nombre_hexa faire
    pour indice allant de 0 à la longueur du tableau tab_hexa faire
        si chiffre = tab_hexa[indice]
            alors mot_binaire = mot_binaire + tab_binaire[indice]
        fin si
    fin pour
fin pour
```

### 1.4. Fonction hexa\_decimal

Elle fait appel aux fonctions hexa\_binaire et binaire\_decimal

## 1.5. Fonction binaire\_hexa

Elle convertit un mot binaire (chaîne de caractères) en nombre hexadécimal (chaîne de caractères).

```

nbre_hexa = vide
nbre = vide
mot = vide
tant que la longueur de mot_binaire ≠ 0 faire
    tant que la longueur de mot_binaire < 4 faire
        mot_binaire = 0 + mot_binaire
    fin tant que
    mot = les 4 bits de poids le plus faible de mot_binaire
    pour indice allant de 0 à la longueur du tableau tab_binaire faire
        si chiffre = tab_binaire[indice]
            alors nbre = nbre + tab_hexa[indice]
        fin si
    fin pour
    mot_binaire = mot_binaire sans les 4 derniers bits de poids le plus faible
fin tant que
pour indice allant de longueur de mot à 0 faire
    nbre_hexa = nbre_hexa + nbre[indice]
fin pour

```

## 1.6. Fonction decimal\_hexa

Elle fait appel aux fonctions decimal\_binaire et binaire\_hexa

## 2. Travail à faire

### 2.1. Programme

- Implémenter et tester chacune des fonctions de conversion.
- **Option** : implémenter une interface pour choisir le type de conversion et le nombre à convertir.

### 2.2. Compte rendu

Fournir un compte rendu individuel, de deux pages maximums, contenant les parties suivantes :

- Objectif du programme
- Difficultés rencontrées
- Méthode de test du programme

## 3. Conditions de travail

Binôme ou seul.

## 4. Travail à rendre

- Fichier Python du convertisseur. Le nom du fichier doit comporter votre nom de famille et être de la forme : **vousre\_nom\_de\_famille\_convertisseur.py**.
- Fichier de compte rendu au format PDF avec un nom de la forme : **vousre\_nom\_de\_famille\_convertisseur.pdf**.

### Grille d'évaluation

Critères	Détails	Points
Respect des consignes	Date d'envoi, nom du fichier...	0,5
Commentaires	Aide à la compréhension du code	1
Lisibilité du code	Nom des variables, nom des fonctions...	0,5
Fonctions	Fonctionnement	3,5
Optimisation du code	Simplification du codage	0,5
Option	Interface	1,5
Compte-rendu	Objectif	0,5
	Difficultés	1
	Méthode	1