

N. S. I. Term. - DS02 (lundi 16 octobre 2023)

Exercice 1.

- 1) Rappeler la valeur des puissances de 2, d'exposant positif compris entre 1 et 10 inclus.
- 2) Quelle est la parité de l'entier naturel dont l'écriture binaire est $(100111101)_2$? Pourquoi ?
- 3) Quels sont les entiers naturels dont l'écriture binaire est $(1000)_2$? $(11101)_2$? $(1111)_2$?
- 4) Déterminer l'écriture binaire des entiers 31, 256 et 157.

On rappelle que : $16^2 = 256$; $16^3 = 4096$; $16^4 = 65536$.

- 5) Quels sont les entiers naturels dont l'écriture hexadécimale est $(e9)_{16}$? $(f0f)_{16}$?
- 6) Déterminer l'écriture hexadécimale des entiers 65540 ? 673 ?

Exercice 2.

1)

On modélise la représentation binaire d'un entier non signé par un tableau d'entiers dont les éléments sont 0 ou 1. Par exemple, le tableau `[1, 0, 1, 0, 0, 1, 1]` représente l'écriture binaire de l'entier dont l'écriture décimale est

$$2^{**6} + 2^{**4} + 2^{**1} + 2^{**0} = 83.$$

À l'aide d'un parcours séquentiel, écrire la fonction `convertir` répondant aux spécifications suivantes :

```
def convertir(T):
    """
    T est un tableau d'entiers, dont les éléments sont 0 ou 1 et
    représentant un entier écrit en binaire. Renvoie l'écriture
    décimale de l'entier positif dont la représentation binaire
    est donnée par le tableau T
    """
```

Exemple :

```
>>> convertir([1, 0, 1, 0, 0, 1, 1])
83
>>> convertir([1, 0, 0, 0, 0, 0, 1, 0])
130
```

2) Quelles modifications apporter au code de la fonction si l'entrée de la fonction est une chaîne de caractères composée de zéros et de uns représentant l'écriture binaire d'un entier positif ?

On pourra écrire un nouveau code pour la fonction ou numéroter les lignes du programme précédent et indiquer les numéros de lignes à modifier et les modifications à apporter.

Exercice 3.

1)

Écrire une fonction `conv_bin` qui prend en paramètre un entier positif `n` et renvoie un couple `(b, bit)` où :

- `b` est une liste d'entiers correspondant à la représentation binaire de `n` ;
- `bit` correspond au nombre de bits qui constituent `b`.

Exemple :

```
>>> conv_bin(9)
([1, 0, 1, 1], 4)
```

Aide :

- l'opérateur `//` donne le quotient de la division euclidienne : `5//2` donne 2 ;
- l'opérateur `%` donne le reste de la division euclidienne : `5%2` donne 1 ;
- `append` est une méthode qui ajoute un élément à une liste existante :

Soit `T=[5, 2, 4]`, alors `T.append(10)` ajoute 10 à la liste `T`. Ainsi, `T` devient `[5, 2, 4, 10]`.

- `reverse` est une méthode qui renverse les éléments d'une liste.

Soit `T=[5, 2, 4, 10]`. Après `T.reverse()`, la liste devient `[10, 4, 2, 5]`.

On remarquera qu'on récupère la représentation binaire d'un entier `n` en partant de la gauche en appliquant successivement les instructions :

```
b = n%2
n = n//2
```

répétées autant que nécessaire.

2) Quelles modifications apporter à la fonction si l'on souhaite que la fonction renvoie cette fois l'écriture binaire de n sous la forme d'une chaîne de caractères ? On pourra écrire un nouveau code pour la fonction ou numéroter les lignes du programme précédent et indiquer les numéros de lignes à modifier et les modifications à apporter.

Exercice 4. Tableaux à deux dimensions

- 1) On considère le tableau à deux dimensions suivant

| | | | |
|---|---|---|---|
| 5 | 3 | 2 | 4 |
| 7 | 1 | 9 | 8 |

Donner la représentation de ce tableau sous la forme d'une liste de listes Python.

- 2) Ce tableau est saisi dans un tableur de la façon suivante :

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 5 | 3 | 2 | 4 | |
| 2 | 7 | 1 | 9 | 8 | |
| 3 | | | | | |

- Si on enregistre tableau au format .csv, quel sera le contenu du fichier ?
 - Par quelle séquence d'instructions peut-on ouvrir et lire ce fichier et en récupérer le contenu dans une variable `contenu`.
Écrire la chaîne de caractère, `contenu`, obtenue.
 - Par quelle séquence d'instructions, utilisant la méthode `.split()`, peut-on transformer la chaîne `contenu` en la liste de listes décrite au 1) ?
- 3) On considère la liste de listes suivante :

`L = [[8, 2, 3], [1, 5, 6], [4, 0, 9]]`

- Par quelle(s) instruction(s) peut-on extraire de L la valeur 6 ?
- Pourquoi peut-on qualifier cette liste de liste de « tableau à deux dimensions » ?
- Par quelle(s) instruction(s) peut-on afficher la première ligne de ce tableau ?
- Par quelle(s) instruction(s) peut-on afficher la dernière colonne de ce tableau ?
- Par quelle(s) instruction(s) peut-on calculer la somme des valeurs de ce tableau ?
- Décrire en langage naturel un algorithme permettant de vérifier qu'une liste de listes représente un tableau à deux dimensions
- Proposer une implémentation de cet algorithme sous la forme d'une fonction `verifie(L)`, renvoyant un booléen indiquant si L est la représentation d'un tableau à deux dimensions ou non.

Exercice 5.

On considère le dictionnaire suivant, représentant les données relatives à un élève :

```
elevel = {'Nom' : 'BERNARD', 'Prénom' : 'Claude', 'Genre' : 'M',
          'Date de naissance' : ''}
```

- Écrire une instruction, utilisant la fonction `print`, pour afficher, en utilisant les données enregistrées dans le dictionnaire `elevel`, le prénom et le nom de cet élève, sous la forme : `Claude BERNARD`.
- Écrire une instruction, pour que la date de naissance de cet élève, non renseignée initialement, soit mise à la valeur 06/12/2007 (sous la forme d'une chaîne de caractères).
- Écrire une instruction pour supprimer du dictionnaire toute information relative au genre de l'élève (clé et valeur).
- Écrire une instruction pour que l'information sur la classe de l'élève soit ajoutée au dictionnaire. L'élève sera supposé être scolarisé-e dans la classe nommée T5.
- À l'aide de quelle(s) instruction(s), grâce à un parcours par les couples (clé, valeur), peut-on afficher toutes les informations sur l'élève.
- Reprendre la question 5 avec un parcours par les clés.

Exercice 6.

On considère la liste de dictionnaires suivante :

```
L = [{'id_eleve': 541, 'DS01': 8.5, 'DS02': 12.5},  
     {'id_eleve': 244, 'DS01': None, 'DS02': 9.5},  
     ...,  
     {'id_eleve': 62, 'DS01': 16.5, 'DS02': 12.5}]
```

donnant les notes des élèves d'une classe dans une matière.

Il est convenu que si un élève était absent lors d'une évaluation, la valeur de la note est `None`.

- 1) Par quelle(s) instruction(s) peut-on construire une liste `Lids` des identifiants des élèves.
- 2) Définir une fonction `absencesDS(L, ds)` prenant en argument une liste ayant la forme de la liste donnée en introduction et une chaîne de caractères donnant le nom du devoir, et renvoyant la liste des identifiants des élèves absents à ce devoir.
- 3) Définir une fonction `absences(L)` prenant en argument une liste ayant la forme de la liste donnée en introduction et renvoyant une liste de couples (identifiant, ds), chaque couple indiquant l'absence d'un élève donné au devoir dont le nom est `ds`.
- 4) Définir une fonction `moy_elv(L, id_eleve)` prenant en argument une liste ayant la forme de la liste donnée en introduction et l'identifiant d'un élève et renvoyant un couple (moyenne, nombre_de_devoirs) donnant la moyenne de l'élève aux devoirs sur lesquels il a été évalué, et le nombre de ces devoirs. La fonction renverra le couple `(None, 0)` si l'élève a été absent à tous les DS.
- 5) Définir une fonction `moyenne(L, ds)` prenant en argument une liste ayant la forme de la liste donnée en introduction et une chaîne de caractères donnant le nom du devoir, et renvoyant la moyenne des élèves à ce devoir.