**Algorithme en pseudo code :**

*loop* 🡨 **Vrai**

**Tant que** *loop* est vrai, **faire**

*enter\_base* 🡨 **Lire**("Base d'entrée: ")

*exit\_base* 🡨 **Lire**("Base de sortie: ")

**Si** *enter\_base* **ou** *exit\_base* est différent de 2, 10 ou 16 **faire**

**Afficher**("Une des bases n'est pas conforme")

**Sinon**:

*loop* 🡨 **False**

*nbr* 🡨 **Lire**("Nombre à convertir en base e*nter\_base*:)

*nbr\_bin* 🡨 "" (rien)

dico 🡨 0 🡨 0000

1 🡨 0001

2 🡨 0010

3 🡨 0011

4 🡨 0100

5 🡨 0101

6 🡨 0110

7 🡨 0111

8 🡨 1000

9 🡨 1001

A 🡨 1010

B 🡨 1011

C 🡨 1100

D 🡨- 1101

E 🡨 1110

F 🡨 1111

**Pour** *i* jusqu'à *'nombre de caractère de nbr'* **faire**

*index* 🡨 *'le caractère en position 'i''*

**Si** *'code ascii décimal de 'index’'* est compris entre 97 (compris) et 122 (compris) **faire**

*index* 🡨 *'le charactère en position' ’code ascii décimal du caractère - 32' dans la table ascii'*

*nbr\_bin* 🡨 *nbr\_bin* + *index*  dans *dico* + ‘espace’

**Sinon Si** *'code ascii décimal de ‘index’'* est compris entre 48 (compris) et 57 (compris) **ou** qu'il est compris entre 65 (compris) et 90 (compris) **faire**

*nbr\_bin* 🡨 *nbr\_bin* + *index* dans *dico* + ‘espace’

**Afficher**("Nombre converti en base 2: ", *nbr\_bin*)

**Notice d’utilisation :**

1. Le programme demande à l’utilisateur de donner la base dans laquelle le nombre donner sera (Base 2, 10 ou 16, sinon le programme repose la question) (Aucun effet pour la suite).
2. Le programme demande à l’utilisateur de donner la base dans laquelle le nombre donner sera converti (Base 2, 10 ou 16, sinon le programme repose la question) (Aucun effet par la suite).
3. Le programme demande à l’utilisateur un nombre en base 16 qui sera converti en base 2.
4. Le programme affiche le nombre converti en base 2
5. Le programme se termine

**Tests :**

|  |  |  |
| --- | --- | --- |
| **Valeur testée :** | **Résultat** **attendu :** | **Résultat :** |
| CA10 | *1100 1010 0001 0000* | **1100 1010 0001 0000** |
| ca10 | *1100 1010 0001 0000* | **1100 1010 0001 0000** |
| CA | *1100 1010* | **1100 1010** |
| ca | *1100 1010* | **1100 1010** |
| 10 | *0001 0000* | **0001 0000** |
| Eft | *Message d’erreur pour prévenir l’utilisateur que ‘T’ n’est pas dans le dictionaire* | **KeyError: 'T'** |

**Bugs rencontrés :**

Key error: '9' (lorsque j'essayai d'accéder au dictionnaire car c'était un la valeur était un str())

Key error: 'A' (parce que je me suis tromper dans les valeurs du dictionnaire)

*dico* = {0: 0000, 1: 0001, ..., 10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E', 15: 'F'}

**Solution:**

*dico* = {'0': '0000', '1': '0001', ... 'A': '1010', 'B': '1011', 'C': '1100', 'D': '1101',

'E': '1110', 'F': '1111'}

ValueError: invalid literal for int() with base 10: 'CA10' (j'ai mis un int() devant le input() du nombre)

**Conclusion :**

Txt