**Algorithme en pseudo code :**

nombre 🡨 **Lire**(« Nombre en base 16 à convertir en base 2 : »)

résultat 🡨 0

exposant 🡨 0

position\_du\_caractère 🡨 -1

**Pour** i allant de 0 à la longueur de nbr **faire**

code\_ascii 🡨 nombre décimal ASCII du dernier caractère de nbr

**Si** le caractère est une lettre majuscule **faire**

code\_ascii 🡨 code\_ascii – 55

résultat 🡨 résultat + code\_ascii \* (16\*\*exposant)

**Sinon si** le caractère est une lettre minuscule **faire**

code\_ascii 🡨 code\_ascii – 87

résultat 🡨 résultat + code\_char \* (16\*\*exposant)

**Sinon faire**

résultat 🡨 résultat + le dernier caractère \* (16\*\*exposant)

exposant 🡨 exposant + 1

position\_du\_caractère 🡨 position\_du\_caractère – 1

binaire 🡨 « »

i 🡨 0

**Tant que** résultat est différent de 0 **faire**

**Si** i est égal à 4 **faire**

Mettre un espace

**Si** résultat est un nombre impair **faire**

Mettre un 1 au nombre binaire final

**Sinon** **faire**

Mettre un 0 au nombre binaire final

résultat 🡨 résultat // 2

i 🡨 i + 1

**Afficher**(Le nombre binaire)

**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* | **1 0000** |
| Eft | *Message d’erreur pour prévenir l’utilisateur que ‘T’ n’est pas compris* | **ValueError: invalid literal for int() with base 10: 't'** |
| (Dans la base d’entrée ou sortie) 17 | *Message pour prévenir qu’une des bases n’est pas conforme* | **Une des bases n'est pas conforme.** |

**Bugs rencontrés :**

Au début on n’a pas transformé la chaine de caractères en entier mais on voulait diviser par 2, ça n’a donc pas marché.  
Quand on demandait le nombre à convertir, la valeur changeait en fonction que ce soit en majuscule ou en minuscule.

**Conclusion :**

Le développement du programme était assez compliqué et nous a pris un peu de temps car on devait faire le programme de deux groupes à la suite.  
J’ai appris beaucoup de nouvelles choses grâce au projet.  
Le programme était compliqué mais on s’est bien amusé en le faisant et en se rendant compte de nos erreurs bêtes.  
Je n’ai pas forcément aimé rédiger le rapport mais ce n’était pas très long donc ça allait. (Hicham)

Le programme était assez amusant à faire ça nous a beaucoup fait réfléchir mais le pseudo-code étant laborieux à rédiger. Je n’ai pas forcément appris de nouvelles choses si ce n’est le travail d’équipe. (Nicolas)

Le projet était cool, comme je m’ennuyais, j’ai fait le code chez moi, mais j’ai utilisé un dictionnaire, qui a été refusé, donc on a recommencé en cours avec les camarades. On a eu un peu de difficulté pour parcourir la chaine de caractère à l’envers mais on a réussi. Même problème avec le rapport qu’on a dû recommencer. Mais je n’ai pas refais le pseudo-code, c’était Nicolas. Au final, j’ai bien aimé le travail. (Leween)