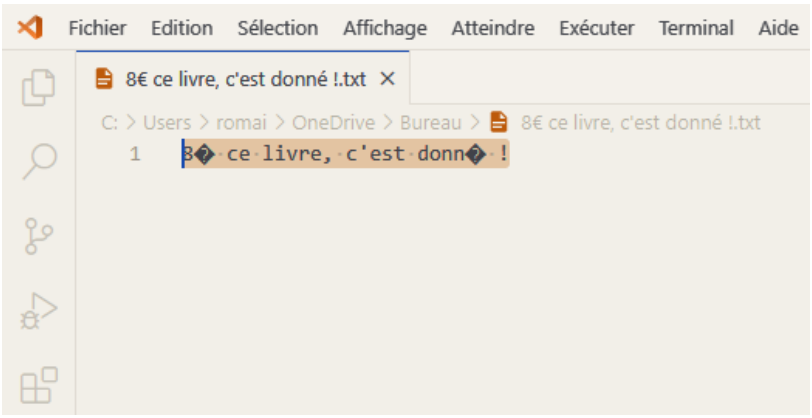
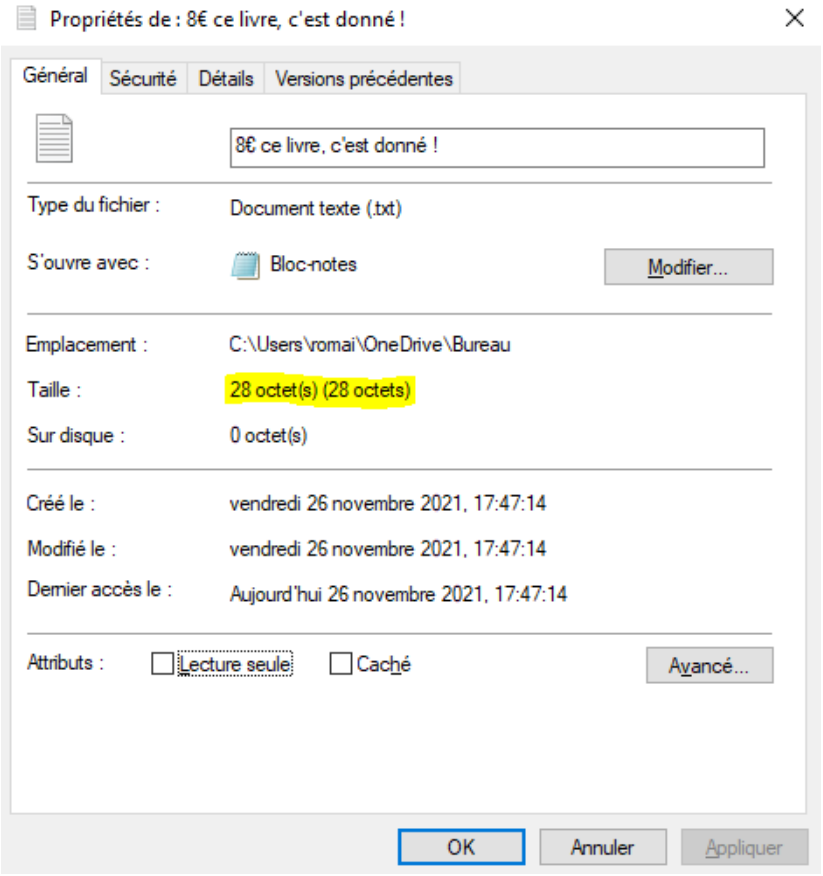
1. Codage d'un texte simple en ISO-LATIN :

Il est impossible d'enregistrer le fichier texte encodé en ISO-8859-1, car ce dernier contient deux caractères qui ne sont pas pris en charge par cette forme d'encodage, il s'agit de : « €; é ».



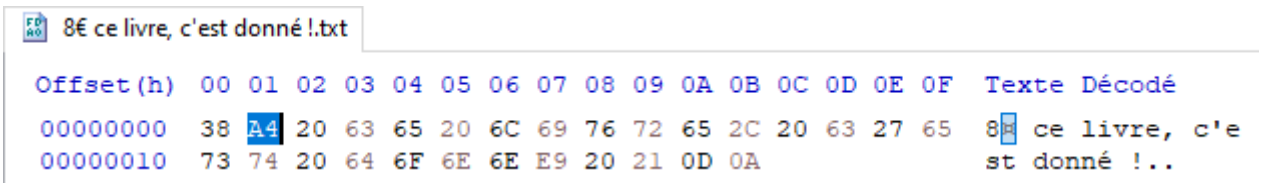
Mon texte est constitué de 26 caractères sur la première ligne.



La taille du fichier texte créé est de 28 octets.

Sur l'éditeur hexadécimal, je constate qu'il y a aussi 28 codes hexadécimaux soit les 28 octets vu précédemment, car il y a bel et bien les 26 caractères de la première ligne mais aussi deux octets supplémentaires dont nous verrons la signification dans la prochaine section : « Les sauts de ligne dans les différents systèmes ».

En ISO-LATIN-15, le caractère « € » est codé par le code hexa « A4 » qui correspond à « ¤ ». (voir image)



Après ces nombreuses études, je peux donc maintenant affirmer que le nombre d'octets nécessaires au codage d'un fichier et égale au nombre de caractères qui le compose mais en rajoutant un certain nombre d'octets pour le retour à la ligne en fonction du système d'exploitation utilisé par l'utilisateur, dans le cas d'un fichier encodé en ISO-LATIN-15.

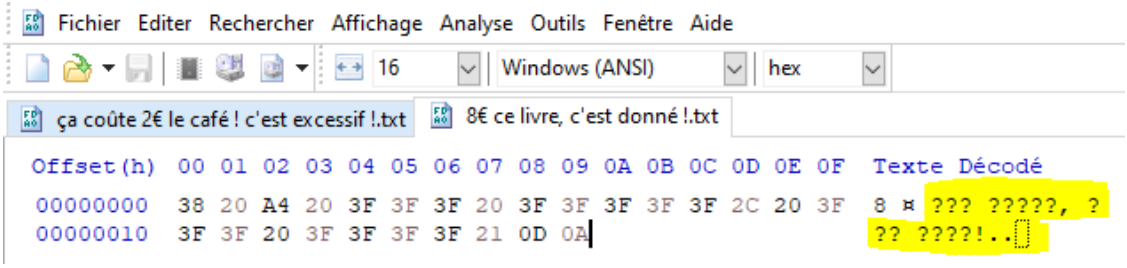
Les sauts de ligne sur les différents systèmes :

Mais attention, sur Windows, si on fait un retour à la ligne à la fin de ce texte alors deux nouveaux octets se rajoutent, il s'agit de « 0D » qui correspond à « Carriage Return » (Retour de chariot) qui définit le retour à la bordure gauche, ainsi que « 0A » qui correspond à « Line Feed » (Saut de ligne) qui définit le passage à la ligne de texte suivante, et qui indique à la machine qu'elle doit être prête à écrire la ligne suivante. Ces deux caractères ne sont donc pas des caractères saisis, mais il rajoute bien deux octets au fichier ! Cette convention est appelée « DOS », elle est utilisée sur les systèmes Windows.

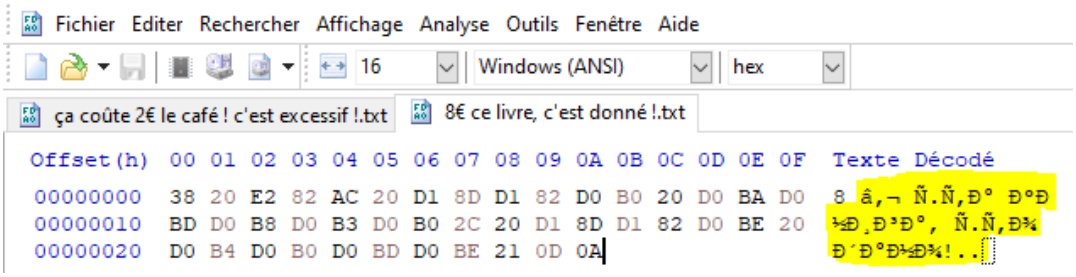
Il existe une deuxième convention appelée « UNIX », elle est utilisée sur les systèmes Linux, Multics, BeOS, AmigaOS ainsi que les nouvelles versions de MacOS (à partir de la version 9). Contrairement à la convention DOS, cette convention permet de définir le saut de ligne par UN seul caractère (« 0A » soit LF) et non pas deux.

Enfin, il existe une troisième convention appelée « MAC » qui était utilisée sur les anciennes versions de MacOS (antérieures à la 9). Contrairement à la convention UNIX, cette convention permet de définir le saut de ligne par le seul caractère (« 0D » soit CR).

En remplaçant « 8 € ce livre, c'est donné ! » par « 8 € эта книга, это дано! » dans l'interpréteur, puis en l'enregistrant encodé en ISO-LATIN-15, des points d'interrogations s'affiche dans HxD à la place des lettres cyrilliques.

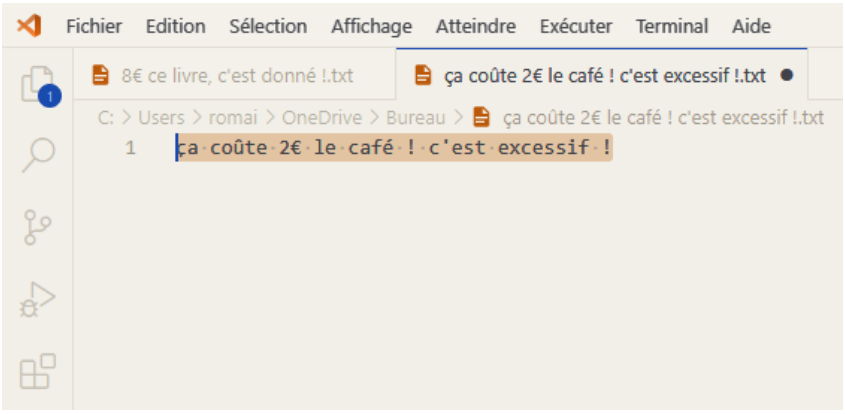


Mais si j'encode ce texte en UTF-8 au lieu de l'ISO-LATIN-15 alors ce ne sont plus des points d'interrogations qui s'affichent à la place des caractères initials mais des caractères totalement différents !

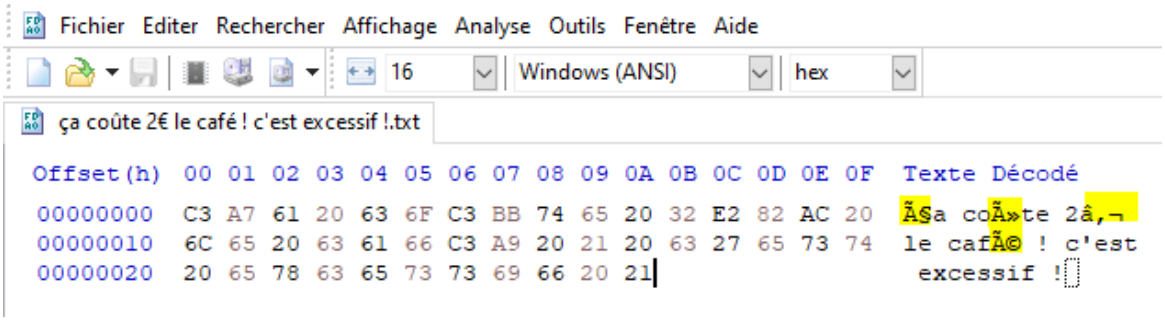


2. Codage d'un texte simple en UNICODE UTF-8 :

Mon texte est constitué de 38 caractères sur la première ligne. Je précise que je n'ai pas fait de retour à la ligne.



Le nombre de codes hexadécimaux contenus dans le fichier est supérieur au nombre de caractères saisis, car les caractères : « ç; û ; € ; é » ont étaient remplacés par plusieurs caractères chacun, ce qui augmente donc le nombre de caractère et par conséquent le nombre d'octets !

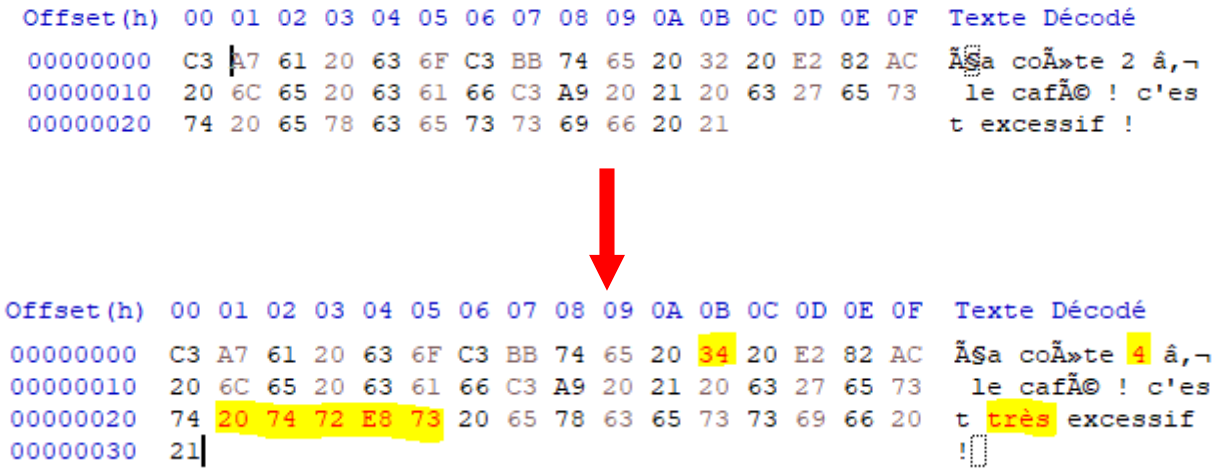


On peut justifier cela en réalisant le tableau suivant :

Caractères	Codes hexadécimaux	Justifications + valeurs Unicode
ç	C3 A7	→ $C3 $A7 → 1100 0011 1010 0111 → 0000 0000 1110 0111 = $E7 → U+00E7 → ç
€	E3 82 AC	→ $E2 $82 $AC → 1110 0010 1000 0010 1010 1100 → 0010 0000 1010 1100 = $20AC → U+20AC → €
û	C3 BB	→ $C3 $BB → 1100 0011 1011 1011 → 0000 0000 1111 1011 = $FB → U+00FB → û
!	21	→ $21 → 0010 0001 → 0010 0001 = $21 → U+0021 → !
É	C3(Ã) 89(‰)	→ $C3 $89 → 1100 0011 1000 1001 → 0000 0000 1100 1001 = $C9 → U+00C9 → É
é	C3(Ã) A9(©)	→ $C3 $A9 → 1100 0011 1010 1001 → 0000 0000 1110 1001 = $E9 → U+00E9 → é

Modification des octets dans un fichier :

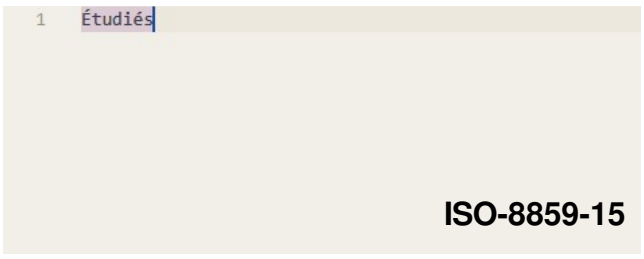
Je dois donc remplacer le « 2 » par un « 4 » et sachant que 4 a pour valeur Unicode : U+0034, je remplace donc la valeur hexadécimale « 32 » par « 34 ». Ensuite, je dois rajouter « très », tout d'abord je rajoute donc un espace en utilisant sa valeur Unicode : U+0020, puis je rajoute le mot en utilisant la valeur Unicode de « t » minuscule (U+0074), la valeur Unicode de « r » minuscule (U+0072), la valeur Unicode de « è » minuscule (U+00E8) et enfin la valeur Unicode de « s » minuscule (U+0073).




```
1   ça coûte 4 € le café ! c'est très excessif !
```

3.

Compatibilité des codes ISO-LATIN et UTF-8 :



Comme démontré juste au-dessus, ce mot étonnant indique que l'expéditeur a utilisé un codage en ISO-LATIN, car il a pu écrire « étudiés » sans souci alors qu'en tant que lecteur je n'arrive pas à lire son message cela indique que j'utilise de mon côté un codage en UTF-8. En regardant de plus près dans l'éditeur hexadécimal on remarque que les caractères : « Ã » et « © » correspondent en réalité au caractère « é », le tableau réalisé précédemment détient cette information.

Les caractères compatibles entre le codage ISO-LATIN et l'UTF-8 sont les caractères non-accentués soit les « Contrôles C0 et latin de base », compris dans la première partie de la table Unicode.