

## SCR.1.1 TP 03 ⊥ :

### Représentation du texte

Les commandes `echo`, `hexdump`, `read`

#### I. Reconstituer un texte à partir de son encodage utf-8.

La commande `hexdump` permet d’afficher le contenu d’un fichier (non nécessairement du texte). Dans cet exercice, il s’agit d’un fichier texte dont on a donné le nom comme argument à `hexdump`. Le résultat de la ligne de commandes correspondante est donné plus loin. Le premier nombre de chaque ligne ne fait pas partie du contenu du fichier. Il indique la position en octet dans le fichier. Le premier octet porte le numéro 0. Sachant qu’on est sur un système d’encodage du texte en UTF-8, reconstituer le texte du fichier. On pourra utiliser le convertisseur à l’adresse <https://unicodelookup.com> en lisant au préalable la section “How-to” qui explique comment entrer une donnée comme argument de la recherche. Quelle est la forme la plus appropriée à soumettre dans le contexte de cet exercice ?

```
00000000  c2 a5 2c 20 e2 99 a0 2c 20 e2 84 97 2c 20 24 2c
00000010  20 c2 a3 2c 20 e2 82 ac 2c e2 98 bb 2c 20 61 72
00000020  65 20 6e 6f 74 20 61 6c 6c 20 63 75 72 72 65 6e
00000030  63 79 20 73 79 6d 62 6f 6c 73 2e 0a
```

#### II. La commande `read` du shell.

Elle permet d’affecter des variables du shell en en saisissant les valeurs à partir de l’entrée standard ou à partir d’un fichier. C’est une commande intégrée au shell.

1. Revoir dans le TP 02 comment faire pour accéder à la documentation d’une commande interne au shell.
2. Revoir dans le TP 02 comment faire pour accéder à la valeur d’une variable du shell.
3. Tester à la ligne de commandes la saisie par `read` de la valeur d’une variable disons `x`.
4. Quelle option donner à `read` pour faire précéder sa saisie par l’affichage à l’écran d’une chaîne de caractères ?
5. Revoir dans le TP 02 quelle forme de lancement utiliser pour soumettre plusieurs commandes sur la même ligne de commandes.
6. Ecrire, sur une même ligne, une ligne de commandes qui affiche un invite à l’utilisateur pour qu’il tape la valeur à placer dans la variable `var` et qui affiche ensuite la valeur de `var`. L’affichage qui invite l’utilisateur doit être fait par la commande `read`.

Dans l’exemple suivant, on tape la ligne de commandes en question et on lui donne une chaîne de caractères que `read` doit saisir dans `var`.

```
Type var value: patatra
var value is: patatra
```

On relance la ligne de commandes en question :

```
Type var value: 495bl stu ...f
var value is: 495bl stu ...f
```

### III. Une commande pour afficher des caractères UNICODE.

1. Quelle option donner à `echo` pour qu'elle interprète les valeurs données en hexadécimal comme les valeurs de caractères UNICODE ? Consulter la documentation appropriée.
2. En utilisant ce qu'on a appris dans la partie précédente, écrire une ligne de commandes qui demande à l'utilisateur de donner une valeur (en hexa), puis affiche le caractère UNICODE correspondant.

Dans l'illustration suivante, on a lancé trois fois la ligne de commandes en question. Les lignes blanches correspondent à l'emplacement de la ligne de commandes en question. On l'a supprimée avant de faire la capture d'écran.

```
Type an HEX UNICODE: 13000
Corresponding CHAR: 𐀀

Type an HEX UNICODE: 130d2
Corresponding CHAR: 𐀍

Type an HEX UNICODE: 1D11F
Corresponding CHAR: 𝄞
```

3. Placer maintenant dans un fichier `print_unicode_char`, la ligne de commandes construite précédemment. Que faut-il faire pour pouvoir exécuter `print_unicode_char` directement comme une commande (cf. TP 01) ? Tester.

### IV. Une commande pour afficher l'encodage utf-8 d'un caractère UNICODE.

1. Quelle option donner à `echo` pour qu'elle ne fasse pas suivre l'affichage par un retour à la ligne ?
2. Consulter la page du manuel de `hexdump` pour voir quelle option on lui a donnée dans l'illustration de la partie I.
3. Par la commande `read` du `shell`, on peut saisir des valeurs dans plusieurs variables à l'aide d'une seule même commande `read`. Tester.
4. Écrire une ligne de commandes qui demande à l'utilisateur de donner une valeur (en hexa), puis affiche la transformation utf-8 du caractère UNICODE correspondant.

Dans l'illustration suivante, on a lancé deux fois la ligne de commandes en question. Les lignes blanches correspondent à l'emplacement de la ligne de commandes en question. On l'a supprimée avant de faire la capture d'écran.

```
Type an HEX UNICODE: 2660
UTF-8 --> e2 99 a0

Type an HEX UNICODE: 1D11F
UTF-8 --> f0 9d 84 9f
```

On pourra vérifier si la ligne de commandes construite fonctionne correctement en comparant son résultat avec les correspondances trouvées dans la partie I., par exemple, ou en cherchant l'encodage utf-8 des caractères dans l'illustration donnée dans la partie III.

5. Placer maintenant dans un fichier `print_utf-8_of_unicode`, la ligne de commandes construite précédemment. On veut l'exécuter directement comme une commande. Tester.