

□ Exercice 1 : un peu de ligne de commande

Dans un dossier `A_ranger` de son répertoire utilisateur, Alice a notamment des fichiers dont les noms se terminent par `.c` ou `.html`. On suppose qu'on se trouve actuellement dans ce dossier.

1. Ecrire les commandes permettant :
 - a) De créer deux dossiers : `LangageC` et `Web`

```
mkdir LangageC Web
```

- b) De déplacer tous les fichiers portant l'extension `.c` dans `LangageC` et tous ceux portant l'extension `.html` dans `Web`

```
mv *.c LangageC
mv *.html Web
```

- c) D'afficher les fichiers restants dans le dossier (y compris les fichiers cachés).

```
ls -a
```

2. Grâce à la commande précédente, Alice s'aperçoit qu'elle a dans ce répertoire un fichier `notes.txt`, quelle commande permet d'afficher le contenu de ce fichier dans le terminal ?

```
cat notes.txt
```

3. Alice souhaite créer un lien physique vers `notes.txt`, ce lien doit se trouver dans son répertoire personnelle dans un dossier existant nommé `Important` et elle veut le nommer `notes.sav`, quelle commande doit-elle taper ?

```
ln notes.txt ~/Important/notes.sav
```

□ Exercice 2

1. Ecrire le prototype d'une fonction `verifie` qui prend en argument un caractère et un entier et renvoie un booléen

```
1 bool verifie(char c, int n)
```

2. On veut utiliser cette fonction depuis le `main` d'un programme

- a) écrire les instructions permettant de déclarer un entier `n` initialisé à 42 et un caractère `c` contenant `@`

```
1 int n = 42;
2 int c = '@';
```

- b) écrire une instruction `printf` permettant d'afficher les valeurs de `n` et de `c` dans le terminal

```
1 printf("n = %d et c = %c",n,c);
```

- c) déclarer un booléen `test` et l'initialiser au résultat de l'appel de `verifie` avec les arguments `c` et `n` définis ci-dessus

```
1 bool test = verifie(c,n)
```

- d) Ecrire une instructions conditionnelle qui affiche « Ok ! » dans le terminal si `test` vaut `true` et « Bug ! » sinon.

```
1 if (test)
2 {
3     printf("Ok !\n");
4 }
5 else
6 {
7     printf("Bug !\n");
8 }
```