

```
cd /home/event/discovery_piscine
mkdir cell03
cd cell03
mkdir ex03
cd ex03
```

```
// DEBUT CREATION DE DOSSIERS ET FICHIERS POUR TEST
```

```
// on crée 38 fichiers et 3 sous-dossiers pour atteindre un compte de 41
```

```
touch file1
touch file2
mkdir dir1
cd dir1
touch file3
touch file4
mkdir dir2
cd dir2
touch file5
touch file6
cd ..
cd ..
mkdir dir3
cd dir3
touch file7
touch file8
cd ..
touch file9 file10 file11 file12 file13 file14 file15 file16 file17 file18 file19 file20 file21 file22 file23
file24 file25 file26 file27 file28 file29 file30 file31 file32 file33 file34 file35 file36 file37 file38
```

```
// FIN CREATION DE DOSSIERS ET FICHIERS POUR TEST
```

```
man find          // manuel pour find
man wc            // manuel pour wc
man chmod         // manuel pour chmod
```

```
find . -mindepth 1 -type f,d | wc -l    // on liste tous les fichiers de type fichier normal (f, regular file) et
```

```
// dossier (d, directory) à partir du répertoire courant (.)
// à partir du 1er niveau de profondeur dans l'arborescence (-mindepth 1)
// on utilise cette option pour ne pas que le répertoire courant (.)
// s'affiche dans le résultat
```

```
// puis on exécute la commande wc -l (word count avec l'option -l,
// pour compter le nombre de lignes de l'entrée fournie)
// (l'entrée fournie peut être un fichier ou le résultat d'une commande)
// sur le résultat de la commande find . -mindepth 1 -type f,d
// (avec | ("pipe") qui redirige la sortie standard
// (le résultat de la commande))
// vers la commande wc -l
// résultat : 41
```

```
// AUTRE SOLUTION :
```

```
find . \! -path . -type f,d | wc -l
```

```
// avec \! -path . on indique qu'on exclut (avec \!)
// le résultat de chemin "." du résultat
// on utilise \ pour échapper le caractère ! afin qu'il soit bien
// interprété par la console
```

```
vim count_files.sh
```

```
i
```

```
#!/bin/bash
```

```
// indique que ce fichier est un script bash et permet
// son interprétation et son exécution
```

```
enter
```

```
find . -mindepth 1 -type f,d | wc -l
```

```
echap
```

```
:wq
```

```
enter
```

```
ls -l
```

```
// on vérifie les droits utilisateur du fichier count_files.sh
```

```
// exemple de résultat : -rw-r--r-- ...
```

```
chmod +x count_files.sh
```

```
// on ajoute les droits d'exécution (+x) au fichier count_files.sh
```

```
// pour qu'on puisse l'exécuter
```

```
ls -l
```

```
// on vérifie les droits utilisateur du fichier count_files.sh
```

```
// exemple de résultat : -rwxr-xr-x ...
```

```

chmod 777 count_files.sh
ls -l

// AUTRE SOLUTION :
// on ajoute tous les droits au fichier count_files.sh
// on vérifie les droits utilisateur du fichier count_files.sh
// exemple de résultat : -rwxrwxrwx ...

./count_files.sh | cat -e

// on exécute le fichier en shell avec ./
// l'option -e de cat indique d'utiliser la notation ^ et M-
// et d'afficher $ à la fin de chaque ligne
// notation ^ : affiche les touches de contrôle
// (ex : Ctrl+C comme ^C, retour chariot comme ^M, etc.)
// notation M- : affiche les touches Meta (ex : Alt+a comme M-a)

// résultat : 42$ (car on compte aussi le fichier count_files.sh !)

rm file*

// supprime tous les fichiers commençant par file
// * : wildcard ("tout ce qui suit")
// file* signifie donc "tout ce qui commence par file"

ls

rm -r dir*
ls

// supprime tous les fichiers commençant par dir

```