

```
cat /etc/passwd
// "affiche la sortie d'un cat /etc/passwd"

// ce fichier contient la liste des comptes du système (1 par ligne)
// avec, pour chacun, le nom d'utilisateur (login), l'ID utilisateur,
// l'ID du groupe, le répertoire de base (dossier home),
// l'interpréteur de commande (shell) utilisé, etc

cat /etc/passwd | grep -v '^#'
// "en retirant les commentaires"
// (les lignes commençant par #)

// on utilise grep '^#'
// pour filtrer les lignes qui commencent par #
// option -v : inverse le sens du matching
// affiche donc les lignes qui ne commencent PAS par #

cat /etc/passwd | grep -v '^#' | sed -n '2~2p'

// "une ligne sur deux, en partant de la seconde"

// avec l'option -n, sed n'affiche aucune ligne du texte passé en entrée,
// sauf celles filtrées
// (le filtre à appliquer est indiqué entre parenthèses)
// on affiche les lignes (p : print = imprimer)
// en partant de la deuxième ligne (2)
// et en sélectionnant chaque 2ème ligne à partir de la position courante
// (~ : tilde, opérateur de séquence numérique) (~2)
// ce filtre sélectionne donc
// une ligne sur deux à partir de la deuxième ligne (incluse)
```

```
cat /etc/passwd | grep -v '^#' | sed -n '2~2p' | awk -F ':' '{ print $1 }' | rev
```

```
// "en inversant chaque login"  
// (le login correspond au premier champ de chaque ligne)  
// on utilise awk -F ':' '{ print $1 }'  
// pour afficher le 1er champ seulement  
// awk est un langage de programmation de manipulation de texte  
// '' délimite le programme à exécuter  
// { } délimite un bloc de commande  
// il contient l'action à exécuter  
// print indique d'afficher ce qui suit à l'écran  
// $1 est une variable intégrée à awk  
// qui représente le premier champ de la ligne d'entrée courante  
  
// par défaut, un champ est séparé du précédent par un espace ou une tabulation  
// l'option -F (field separator) permet de définir un autre séparateur de champ  
// chaque champ du fichier /etc/passwd est séparé par ":"  
// on indique donc que le séparateur des champs est ":"  
// avec -F ':'  
  
// rev (reverse) : inverse chacun des caractères  
// dans la (ou les) ligne(s) fournies en entrée
```

```
cat /etc/passwd | grep -v '^#' | sed -n '2~2p' | awk -F ':' '{ print $1 }' | rev | sort -r
```

```
// "en triant par ordre alphabétique inversé"  
// sort : trie par ordre alphabétique  
// option -r (reverse) : inverse le résultat  
// (trie donc par ordre alphabétique inverse)
```

```
export FT_LINE1=7          // on initialise la variable d'environnement FT_LINE1 à 7  
export FT_LINE2=15         // on initialise la variable d'environnement FT_LINE1 à 15
```

```
cat /etc/passwd | grep -v '^#' | sed -n '2~2p' | awk -F ':' '{ print $1 }' | rev | sort -r | sed -n  
"$FT_LINE1,$FT_LINE2 p"
```

```
// "en ne conservant que les logins compris entre FT_LINE1 et FT_LINE2 inclus"  
// avec l'option -n, sed n'affiche aucune ligne du texte passé en entrée,  
// sauf celles filtrées  
// (le filtre à appliquer est indiqué entre parenthèses)  
// on affiche les lignes (p : print = imprimer)  
// en partant de la ligne numéro $FT_LINE1  
// jusqu'à la ligne numéro $FT_LINE2  
// ($FT_LINE1 correspondant à la valeur de la variable d'environnement FT_LINE1,  
// idem pour $FT_LINE2)
```

```
cat /etc/passwd | grep -v '^#' | sed -n '2~2p' | awk -F ':' '{ print $1 }' | rev | sort -r | sed -n  
"$FT_LINE1,$FT_LINE2 p" | sed -z 's/\n/, /g'
```

```
// "séparés par des ", " (sans les guillemets)"
```

```
// ATTENTION : le résultat est affiché sur plusieurs lignes  
// il faut donc tout d'abord concaténer toutes ces lignes  
// pour pouvoir travailler sur celles-ci (les formater dans leur ensemble)  
// avec l'option -z, sed concatène les différentes lignes  
// et ajoute un saut de ligne (\n) à la fin de chacune d'elle
```

```
// s : substitution  
// utilisé avec sed pour remplacer un premier motif (entre le premier et le deuxième slash)  
// avec un deuxième motif (entre le deuxième et le troisième slash)  
// on remplace donc ici \n (le saut de ligne) par ', ' (sans les guillemets)
```

```
// g : global  
// utilisé avec sed pour remplacer toutes les occurrences de \n par ', ' (sans les guillemets)  
// sinon, seulement la première occurrence de \n sera remplacée par ', ' (sans les guillemets)
```

```
cat /etc/passwd | grep -v '^#' | sed -n '2~2p' | awk -F ':' '{ print $1 }' | rev | sort -r | sed -n  
'$FT_LINE1','$FT_LINE2'p' | sed -z 's/\n/, /g' | sed 's/, $/./'
```

```
// "et terminés par un "."  
// on utilise de nouveau (avec s) avec sed  
// pour remplacer la dernière($) virgule suivie d'un espace (" ", " (sans les guillemets))  
// par un point (.)
```

```
vim r_dwssap.sh
```

```
<i>
```

```
#!/bin/sh
```

```
cat /etc/passwd | grep -v '^#' | sed -n '2~2p' | awk -F ':' '{ print $1 }' | rev | sort -r | sed -n  
"$FT_LINE1,$FT_LINE2 p" | sed -z 's/\n/, /g' | sed 's/, $/./'
```

```
<escape>
```

```
:wq
```

```
<enter>
```

```
chmod +x r_dwssap.sh
```

```
./r_dwssap.sh
```

```
// Résultat : spoonrek, semag, repleh-kp-spuc, rehctapsid-hceeps, pukcab, pmudpct, pl, pilph, mpvnepo-mn.
```

```
// ASTUCES :  
// man grep  
// man sed  
// man awk  
// man rev  
// man sort
```