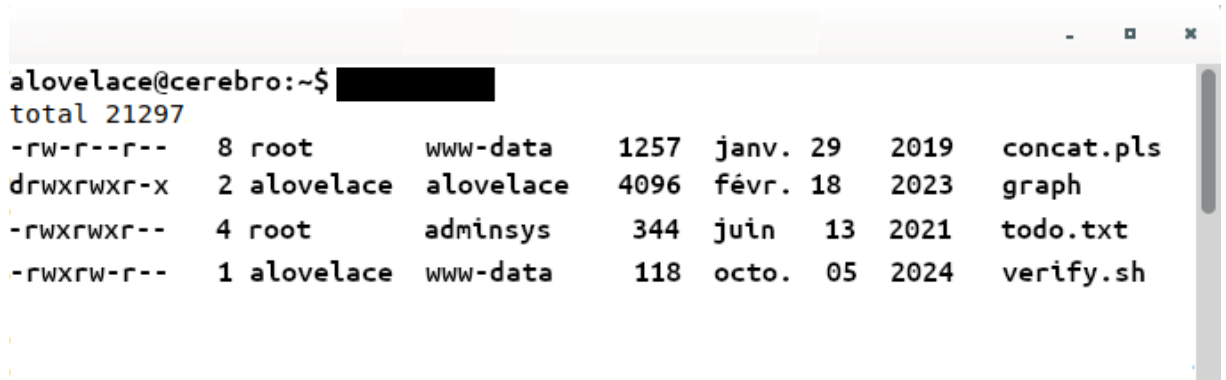


Exercice 2 (6 points)

Cet exercice porte sur les systèmes d'exploitation ainsi que sur programmation et la programmation orientée objet.

Partie A

1. Écrire, dans un système Linux, la commande masquée en noir qui a permis d'obtenir le résultat de la Figure 1.



```
alovelace@cerebro:~$ XXXXXXXXXX
total 21297
-rw-r--r--  8 root      www-data    1257  janv. 29  2019  concat.pls
drwxrwxr-x  2 avelace  avelace    4096  févr. 18  2023  graph
-rwxrwxr--  4 root      adminsys    344   juin  13  2021  todo.txt
-rwxrw-r--  1 avelace  www-data    118   octo.  05  2024  verify.sh
```

Figure 1. Capture d'écran d'un terminal Linux

Dans la Figure 1, les permissions des fichiers apparaissent au début de chaque ligne et sont composées de 10 caractères. En lisant de la gauche vers la droite :

- le premier caractère indique la nature du fichier, `-` pour un fichier classique, `d` pour un répertoire et `l` pour un lien ;
 - ensuite, on trouve trois groupes de 3 caractères chacun, indiquant si le fichier (ou répertoire) est autorisé en lecture `r`, écriture `w` ou exécution `x`. Si la permission n'est pas accordée, le caractère est remplacé par `-`. Les 3 groupes correspondent, dans cet ordre, aux droits du propriétaire (user), du groupe (group) puis du reste des utilisateurs (other).
2. Décrire les permissions que les trois types d'utilisateurs (user, group ou other) ont sur le fichier `verify.sh` d'après la Figure 1.
 3. D'après la Figure 1, écrire la commande que l'utilisateur, propriétaire du fichier `concat.pls`, doit saisir pour supprimer ce fichier.

On s'intéresse à la façon de modifier les permissions d'un fichier avec la commande `chmod`. Pour cela on saisit dans un terminal la commande `man chmod`. La Figure 2 ci-après reproduit l'affichage obtenu.

`chmod` - Modifier les bits de comportement de fichier

SYNOPSIS

```
chmod [OPTION]... MODE[,MODE]... FICHIER...
chmod [OPTION]... MODE-OCTAL FICHIER
chmod [OPTION]... --reference=FICHIER-R FICHIER
```

Cette page de manuel documente la version GNU du programme `chmod`. Le programme `chmod` modifie les bits de comportement de fichier de chacun des fichiers indiqués, en suivant l'indication de mode, qui peut être une représentation symbolique des changements à effectuer, ou un nombre octal représentant le motif binaire des nouveaux bits de comportement.

Le format d'un mode symbolique `[ugoa...][[+|=][permissions...]]...`, où `permissions` vaut soit zéro, soit plusieurs lettres de l'ensemble `rwX`, soit une seule lettre de l'ensemble `ugo`. Plusieurs modes symboliques peuvent être indiqués ensemble, séparés par des virgules.

Une combinaison des lettres `ugoa` contrôle la catégorie d'accès à modifier. Il peut s'agir de l'utilisateur possédant le fichier (`u`), des autres utilisateurs du même groupe que le fichier (`g`), des utilisateurs n'appartenant pas au groupe du fichier (`o`), ou de tous les utilisateurs (`a`).

L'opérateur `+` ajoute à chaque fichier les bits de comportement de fichier spécifiés à ceux déjà existants, l'opérateur `-` les enlève,

Les lettres `rwX` sélectionnent les bits de comportement de fichier des utilisateurs concernés : lecture (`r`), écriture (`w`), exécution (ou recherche pour les répertoires) (`x`).

Figure 2. Extrait de la page de manuel de `chmod`

4. Avec les éléments de la Figure 1 et à la lecture de la documentation Figure 2, expliquer quel sera l'effet de l'instruction `chmod g-wx todo.txt`.

Partie B

Dans la suite de la documentation de la commande `chmod`, on peut lire l'extrait suivant.

l'indication de mode, qui peut être une représentation symbolique des changements à effectuer, ou un nombre octal représentant le motif binaire des nouveaux bits de comportement.

Cela signifie que l'on peut indiquer les permissions soit sous la forme d'une chaîne de 9 caractères composée de `-`, `r`, `w` et `x`, appelée *représentation symbolique*, soit sous la forme d'une valeur numérique `abc` composée de trois chiffres `a`, `b` et `c` compris entre 0 et 7 appelée *représentation octale*.

Le but de cette partie est d'écrire des fonctions Python permettant de passer d'une représentation symbolique à une représentation octale des permissions.

On considère une chaîne de trois caractères dont :

- le premier caractère est '-' ou 'r' ;
- le second caractère est '-' ou 'w' ;
- et le dernier est '-' ou 'x'.

On souhaite déterminer le motif binaire de 3 bits correspondant à cette chaîne. Pour cela, on applique les règles suivantes :

- si le premier caractère est 'r', le premier bit (celui le plus à gauche), vaut 1, sinon il vaut 0 ;
 - si le second caractère est 'w', le second bit vaut 1, sinon il vaut 0 ;
 - Si le dernier caractère est 'x', le dernier bit vaut 1, sinon il vaut 0.
5. Déterminer l'écriture en base 10 du motif binaire de 3 bits correspondant à la chaîne 'rw-'.

La représentation octale associée à une représentation symbolique est obtenue en appliquant ce qui précède trois fois :

- les trois caractères correspondant à user donnent le chiffre de gauche ;
- les trois suivants (correspondant à group) donnent le chiffre du milieu ;
- les trois derniers (correspondant à other) donnent le chiffre de droite.

Par exemple, la représentation octale associée à la représentation symbolique `rwxr--` est 754.

6. Écrire le code d'une fonction Python `bin_to_oct` qui prend en paramètre une chaîne de 3 caractères composée uniquement des caractères '0' et '1' et renvoie la représentation octale correspondant à ce motif binaire (le bit de poids fort étant encore à gauche).

Exemples :

```
>>> bin_to_oct('101')
5
>>> bin_to_oct('011')
3
```

7. On considère la fonction Python `mystere` qui prend en paramètre une chaîne de 9 caractères et qui renvoie une chaîne de caractères.

```
1 def mystere(chaine):
2     resultat = ''
```

```

3  for c in chaine:
4      if c == '-':
5          resultat = resultat + '0'
6      else:
7          resultat = resultat + '1'
8  return resultat

```

Déterminer ce que renvoie `mystere('rw-r-xr--')`.

8. On considère dans cette question qu'on dispose d'une fonction Python `symb_to_bin` qui prend en paramètre une chaîne de 9 caractères correspondant à l'écriture symbolique d'une permission et qui renvoie une chaîne de caractères correspondant au motif binaire de cette permission.

Recopier et compléter le code de la fonction Python `symb_to_oct` donné ci-dessous, qui prend en paramètre une représentation symbolique sous la forme d'une chaîne de caractères, et qui renvoie la représentation octale correspondante sous la forme d'une chaîne de caractères. Il est possible de répondre en une ou plusieurs lignes de code.

```

1 def symb_to_oct(chaine):
2     repr_bin = symb_to_bin(chaine)
3     # les 3 premiers bits correspondent au propriétaire
4     user = repr_bin[0] + repr_bin[1] + repr_bin[2]
5     # les 3 suivants au groupe
6     group = repr_bin[3] + repr_bin[4] + repr_bin[5]
7     # et les 3 derniers aux autres
8     other = repr_bin[6] + repr_bin[7] + repr_bin[8]
9     ...

```

Exemples :

```

>>> symb_to_oct('rwxrw-r--')
'764'
>>> symb_to_oct('rw-r--r--')
'644'

```

Partie C

Dans cette partie, on souhaite écrire un gestionnaire de fichiers Linux. Pour cela, on modélise les fichiers en utilisant la programmation orientée objet. On dispose de la classe `Fichier` ci-dessous.

```

1 class Fichier:
2     def __init__(self, nom, poids, proprio, groupe,
permission):
3         self.nom = nom
4         self.poids = poids           # taille en octets
5         self.proprietaire = proprio # nom du propriétaire
6         self.groupe = groupe        # nom du groupe
7         self.permission = permission # représentation symbolique

```

9. Instancier l'objet `mon_fichier` de la classe `Fichier` représentant un fichier nommé `concat.pls`, dont le propriétaire est `root`, le groupe est `www-data`, le poids 1257 octets et dont les permissions sont données par la représentation symbolique `'rw-r--r--'`.
10. Écrire une méthode `chown` de la classe `Fichier` qui prend en paramètre une chaîne de caractères qui correspond au nom du nouveau propriétaire et qui modifie le propriétaire du fichier.
11. Écrire une fonction `get_executable` qui prend en paramètres une liste d'objets de la classe `Fichier` ainsi qu'un nom de propriétaire et qui renvoie la liste des fichiers qui appartiennent à ce propriétaire et qui sont exécutables par ce propriétaire.