Exercise 1:

1. How is permission granted to delete a file?

Permission is granted to delete a file by having both write 'w' and execute 'x' permissions on its parent directory so the file itself does not control the deletion.

2. to create a file that my colleague can edit but not delete :

We first create a file using the command : touch file.txt

Then we give write permission for that file either to file group or all other member by the command:

chmod 664 file.txt (write permission given to group) | chmod 646 (write permission given to all other users)

Note: in case of granting write permission by group, the user who want to edit the file should belong to the same group of the file.

And to prevent it from deleting the file we should revoke write permission of group and other users form the parent directory by the command: chmod 755 parent_dir.

3. **to create a file that my colleague can delete but not edit :**

We first create a file using the command : touch file2.txt

Then we give write and execute permissions for the parent directory of the file either to group or all other member by the command: chmod 777 file2_parent_dir

And to prevent him from editing the file we need to revoke the write permission from the file using the command:

chmod 755 file2.txt

4**. Does it make sense to be able to assign such rights? What are the practical consequences of this experience?**

Assigning such rights is essential for maintaining a secure and stable computing environment. It allows administrators to balance usability and security, ensuring that users can do their tasks without affecting the system's integrity.