1. You have a file with permissions -rw-r--r--, and you run chmod +x file.sh. What happens?
The file initially has read and write access for the owner and read-only for others. Running chmod +x file.sh adds execute permission to all, changing the permissions to -rwxr-xr-x. Now, the file is executable by everyone.

2. What is the difference between chmod 744 file.txt and chmod u=rwx,go=r file.txt?
Both commands assign the same permissions (-rwxr--r--). The difference lies in style—744 is numeric (octal), whereas u=rwx,go=r is symbolic, making it more descriptive and easier to understand at a glance.

3. What is the sticky bit, and when should you use it?
A sticky bit restricts deletion or renaming of files within a directory to the file's owner or root, even if others have write access. It's ideal for shared directories like /tmp.
chmod +t bench.py
ls -ld
includes 't' → drwxrwxrwt

4. You are told to give the owner full access, group only execute, and others no permissions. What symbolic command achieves this?
Use the command:
chmod u=rwx,g=x,o= bench.py
This grants all permissions to the owner, execute-only to the group, and denies all access to others.

5. What is umask, and why is it important?
umask controls the default permissions when files or directories are created. It subtracts permission bits from the default (666 for files, 777 for directories).
umask 0077
File → 600 (rw-------)
Dir → 700 (rwx------)

6. If the umask is 022, what are the default permissions for a new file and a new directory?

File default: 666 - 022 = 644 → rw-r--r--
Directory default: 777 - 022 = 755 → rwxr-xr-x
So, the owner gets full rights, group and others get read access (files) or read/execute (directories).

7. Why is umask often set to 002 in development environments but 027 or 077 in production?

002 (Dev): Files 664, Dirs 775 → Promotes collaboration within the group.
027 (Prod): Files 640, Dirs 750 → Restricts group to read-only; no access for others.
077 (Secure): Files 600, Dirs 700 → Only owner has access, ensuring maximum privacy.

8. UserAdd vs AddUser

useradd: A basic, low-level tool that works across all Linux systems. It requires flags to set things like home directory or shell, making it script-friendly.
adduser: A higher-level command (mainly in Debian-based distros) that walks you through user creation interactively, asking for details like password and name.