Program 6.b. : Perform the following execution using find command

i. Find all the files in a current directory, whose permissions are 0777.

ii. Assign a sticky bit to all the files in a current directory.

iii. Find Directories with full permissions (777) and by using chmod command change the permissions by assigning read, write and execute permissions to owner and only read & execute to group and others.

iv. Find last 20 days modified files, accessed files.

v. Find all the files which are modified in last 1 hour.

Solution:

i. Find all the files in the current directory with permissions set to 0777:
find . -type f -perm 0777

- `.`: Represents the current directory.
- `-type f`: Specifies that only files should be considered.
- `-perm 0777`: Filters files with permissions set to 0777.

ii. Assign a sticky bit to all the files in the current directory:
find . -type f -exec chmod +t {} +

- `.`: Represents the current directory.
- `-type f`: Specifies that only files should be considered.
- `-exec chmod +t {} +`: Executes `chmod +t` on each file found, assigning the sticky bit.

iii. Find directories with full permissions (777) and change the permissions as described:
find . -type d -perm 777 -exec chmod 755 {} +

- `.`: Represents the current directory.
- `-type d`: Specifies that only directories should be considered.
- `-perm 777`: Filters directories with permissions set to 777.
- `-exec chmod 755 {} +`: Executes `chmod 755` on each directory found, modifying the permissions.

iv. Find files modified in the last 20 days and files accessed in the last 20 days:
find . -type f -mtime -20 # Modified in the last 20 days
find . -type f -atime -20 # Accessed in the last 20 days

- `.`: Represents the current directory.
- `-type f`: Specifies that only files should be considered.
- `-mtime -20`: Filters files modified in the last 20 days.
- `-atime -20`: Filters files accessed in the last 20 days.

v. Find all files modified in the last 1 hour:
find . -type f -mmin -60

- `.`: Represents the current directory.
- `-type f`: Specifies that only files should be considered.
- `-mmin -60`: Filters files modified in the last 60 minutes (1 hour).

These `find` commands will help you achieve the specified tasks in your program.

What is permissions set to 0777:

In Linux and Unix-like operating systems, file and directory permissions are represented using a three-digit octal (base-8) notation. Each digit in the octal notation corresponds to a specific permission group:

- **First Digit (Owner):** Represents permissions for the owner of the file or directory.
- **Second Digit (Group):** Represents permissions for the group associated with the file or directory.
- **Third Digit (Others):** Represents permissions for others (users who are not the owner and not in the group).

Each digit is a combination of three bits, and the permission bits are represented as follows:

- **Read (r):** 4
- **Write (w):** 2
- **Execute (x):** 1

The octal number is calculated by adding the values of the permission bits. For example:

- `rwx` (read, write, execute) is represented as `7` (4 + 2 + 1).
- `rw-` (read, write, no execute) is represented as `6` (4 + 2).
- `r--` (read, no write, no execute) is represented as `4` (4).

Now, when permissions are set to `0777`, it means:

- **Owner (User):** `rwx` (4 + 2 + 1)
- **Group:** `rwx` (4 + 2 + 1)
- **Others:** `rwx` (4 + 2 + 1)

In other words, the file or directory has full read, write, and execute permissions for the owner, the group, and others. This is the most permissive setting, allowing anyone to read, write, and execute the file or directory. However, using `0777` permissions should be done with caution, as it may pose security risks by giving unrestricted access to the file or directory. It's generally recommended to use more restrictive permissions based on the specific needs of the file or directory.

What is Assign a sticky bit to all the files in a current directory.

The sticky bit is a permission in Unix and Unix-like operating systems that can be set on directories. When the sticky bit is set on a directory, it has a specific meaning: only the owner of a file within that directory can delete or rename the file, even if other users have write permissions on the same directory.

To assign the sticky bit to all the files in the current directory, you can use the **`chmod`** command in combination with the **`find`** command. Here's the command:

```bash
find . -type f -exec chmod +t {} +
```

Let's break down the command:

- **`.`**: Represents the current directory.
- **`-type f`**: Specifies that only files should be considered (not directories).
- **`-exec chmod +t {} +`**: Executes the **`chmod +t`** command on each file found. The **`+t`** option sets the sticky bit.

This command finds all files in the current directory and its subdirectories and assigns the sticky bit to each file. The sticky bit ensures that only the file owner can delete or rename their own files within that directory.

It's worth noting that the sticky bit is more commonly used on directories (e.g., **`/tmp`**), and assigning it to individual files might not have as common use cases as with directories.