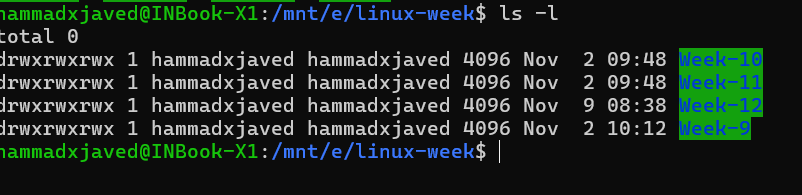
Week 8

**1. Display detailed information about files using ls -l and other ls options.**

**Display Detailed Information About Files:**

* **ls -l**: Lists files in the current directory with detailed information.

ls -l



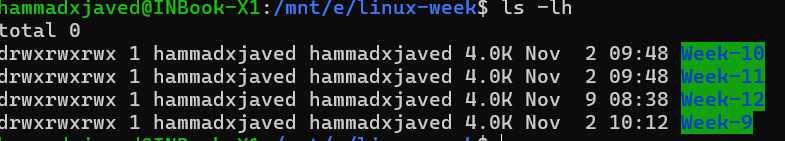
* **ls -a**: Lists all files, including hidden files.

ls -a



* **ls -lh**: Lists files with human-readable file sizes.

ls -lh



**2. Change file ownership using chown and group ownership using chgrp.**

**Change File Ownership:**

* **chown**: Changes the owner of a file.

chown new\_owner file\_name

* **chgrp**: Changes the group ownership of a file.

chgrp new\_group file\_name

3. Modify file and directory permissions using chmod.

**Modify File and Directory Permissions:**

* **chmod**: Changes the permissions of a file or directory.

chmod permissions file\_name

For example, to give the owner read, write, and execute permissions, and others read-only permissions:

chmod 755 file\_name

4. Demonstrate how to set and remove directory permissions.

**Set and Remove Directory Permissions:**

* To set directory permissions:

chmod permissions directory\_name

For example, to give the owner read, write, and execute permissions, and others read and execute permissions:

chmod 755 directory\_name

* To remove directory permissions:

chmod permissions directory\_name

For example, to remove write permissions for others:

chmod o-w directory\_name

5. Write a python program that reads a string which contains English alphabets and numbers. The program should create two lists L1 and L2, where L1 includes all the numbers present in the string while L2 includes all the alphabets of the string.

def separate\_numbers\_and\_alphabets(input\_string):

    L1 = []

    L2 = []

    for char in input\_string:

        if char.isdigit():

            L1.append(char)

        elif char.isalpha():

            L2.append(char)

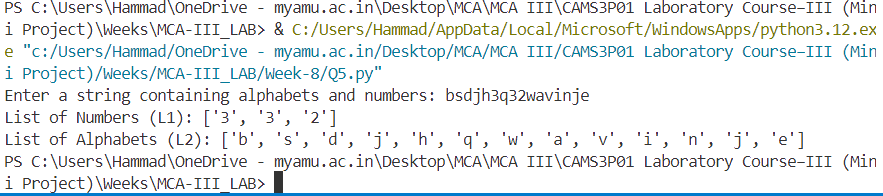
    return L1, L2

input\_string = input("Enter a string containing alphabets and numbers: ")

numbers, alphabets = separate\_numbers\_and\_alphabets(input\_string)

print("List of Numbers (L1):", numbers)

print("List of Alphabets (L2):", alphabets)



6. Write a program in python to find vowels having maximum number of instances in a given file. (Note: File contains variety of data such as English alphabets, numbers etc.).

def find\_max\_vowel\_in\_file(file\_path):

    vowels = 'aeiou'

    vowel\_count = {v: 0 for v in vowels}

    try:

        with open(file\_path, 'r') as file:

            content = file.read().lower()

            for char in content:

                if char in vowels:

                    vowel\_count[char] += 1

        max\_vowel = max(vowel\_count, key=vowel\_count.get)

        return max\_vowel, vowel\_count[max\_vowel]

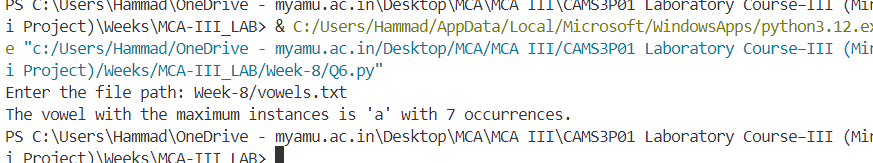
    except FileNotFoundError:

        return "File not found."

file\_path = input("Enter the file path: ")

max\_vowel, count = find\_max\_vowel\_in\_file(file\_path)

print(f"The vowel with the maximum instances is '{max\_vowel}' with {count} occurrences.")



7. Write a Python program to create a list of user’s supplied distinct integers having even number of elements. The program further creates two lists of equal lengths based on the original list, where first list is having all elements less than elements of second list. Display both the lists.

def create\_even\_split\_lists():

    input\_list = input("Enter distinct integers separated by spaces: ")

    integers = list(map(int, input\_list.split()))

    if len(integers) % 2 != 0:

        print("The list must have an even number of elements.")

        return

    integers.sort()

    mid = len(integers) // 2

    L1 = integers[:mid]

    L2 = integers[mid:]

    less\_than\_list = [x for x in L1 if x < L2[0]]

    greater\_than\_list = [x for x in L2 if x > L1[-1]]

    print("List 1 (L1):", less\_than\_list)

    print("List 2 (L2):", greater\_than\_list)

create\_even\_split\_lists()

