Name: Nivedita Londhe

PRN: 22420003

Batch: EN-4

Practical No. 6

Title: Execution of Python file in Linux

PART A:

Single file Execution

➢ **Step 1: Created a new Python** (.py) **file**

I used the nano text editor to create a file named add.py

Command: nano add.py

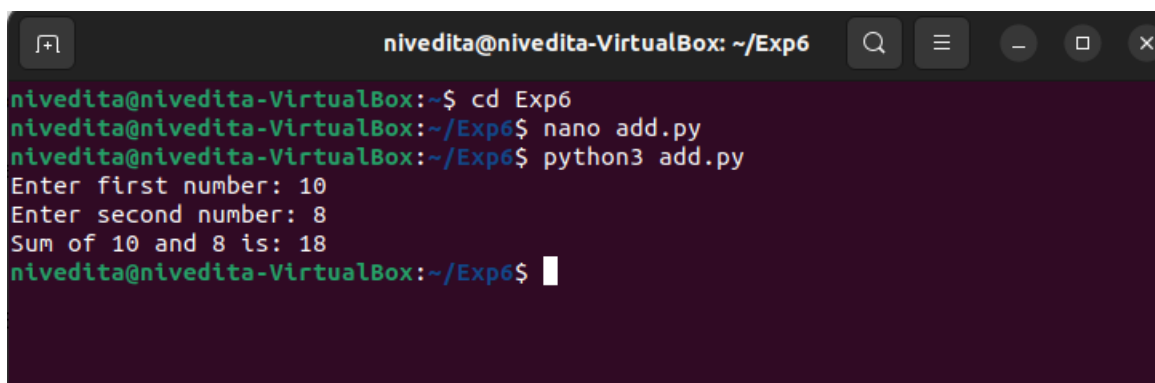➢ **Step 2: Wrote the Python code**

I entered the following code into the file and saved that.

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
sum = a + b
print(f"Sum of {a} and {b} is: {sum}")
```

➢ **Step 3: Run the Python file**

Execute the program using Python 3

Command: python3 add.py

PART B:

Multiple file Execution

➢ **Step 1: Create Source Files**

1. Create separate .py files for operation: loop.py, conditional.py,

   Command: nano file.py

   sub.py

```python
x = int(input("Enter first number: "))
y = int(input("Enter second number: "))
difference = x - y
print(f"Difference between {x} and {y} is: {difference}")
```

   conditional.py

```python
num = int(input("Enter a number: "))
if num % 2 == 0:
    print(f"{num} is even")
else:
    print(f"{num} is odd")
```

   loop.py

```python
n = int(input("Enter a number to print the range: "))
for i in range(n):
    print(f"Number: {i}")
```

➢ **Step 2: Using Shell Script to Run Multiple Files**

1. Create a Shell Script (runPython.sh):

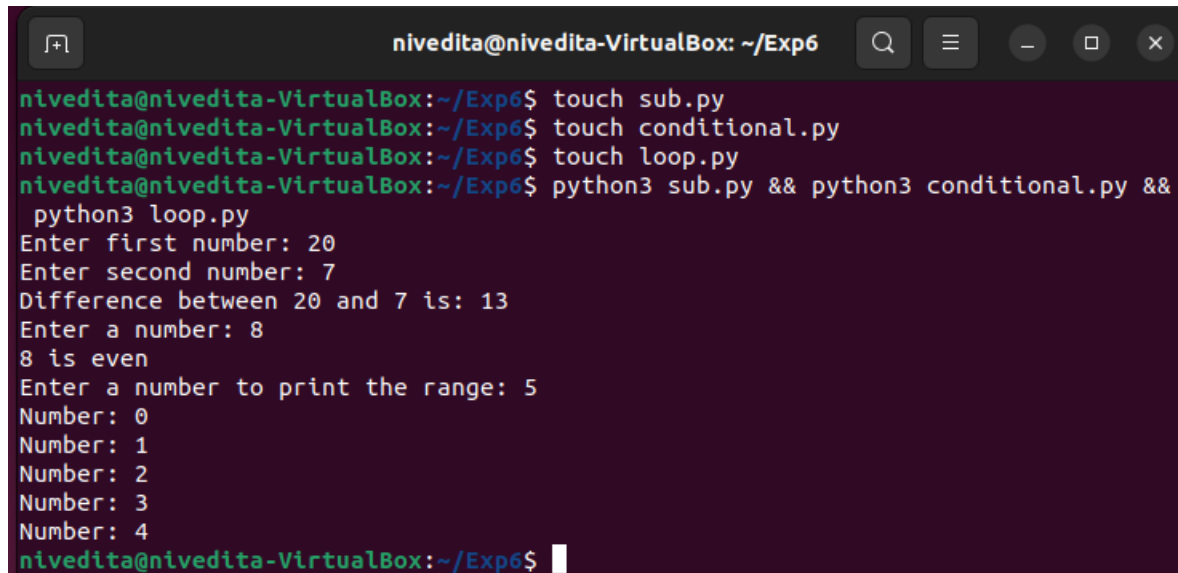   **python3 add.py && python3 sub.py && python3 conditional.py && python loop.py**

2. **commands**:

   Make the shell script executable by running: **chmod +x run_files.sh**

   Run the shell script: **./run_files.sh**

➤ Step 3: Using && Operation to Execute Multiple Files

Run all Python files sequentially in one line

Conclusion:

Executing Python files in Linux can be done easily using the python3 command. For a single file, it runs directly with user input and displays results. Multiple files can be executed using a shell script or the && operator, running them sequentially. Adding user input makes the scripts interactive, allowing dynamic calculations and outputs. This method simplifies batch execution of Python programs in a Linux environment.

Questions:

1. **Write meaning and use of GCC in Linux.**

➔ In Linux, an interpreter (like Python) executes code line by line, translating each instruction into machine code on the fly. This allows for immediate execution, making it useful for scripting, testing, and dynamic languages. However, interpreted programs tend to run slower since each line is processed during runtime.

A compiler (like GCC for C/C++) translates the entire source code into machine language in one step before execution, generating an executable file. This approach leads to faster execution at runtime but requires the additional compilation step beforehand. Both interpreters and compilers are widely used in Linux, depending on the programming language.