CDAC MUMBAI

**Concepts of Operating System Assignment 2**

**Part A**

**What will the following commands do?**

* echo "Hello, World!"

This command prints the text "Hello, World!" to the terminal.

* name="Productive"

This assigns the value "Productive" to a variable named name. You can reference it later using $name.

* touch file.txt

Creates an empty file named file.txt if it doesn't already exist. If it does exist, touch updates the file's last modified timestamp.

* ls -a  
  Lists all files and directories in the current directory, including hidden ones
* rm file.txt  
  Deletes the file named file.txt.
* cp file1.txt file2.txt  
  Copies the contents of file1.txt to file2.txt. If file2.txt doesn't exist, it's created.
* mv file.txt /path/to/directory/  
  Moves file.txt to the specified directory. This can also be used to rename files.
* chmod 755 script.sh  
  Changes the permissions of the file script.sh to 755, which means:

Owner: read, write, and execute (7).

Group: read and execute (5).

Others: read and execute (5).

* grep "pattern" file.txt  
  Searches for the string "pattern" in file.txt and prints any matching lines.
* kill PID  
  Sends a termination signal to the process with the specified Process ID (PID), effectively stopping it.
* mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt  
  Creates a directory named mydir.

Changes into the mydir directory.

Creates an empty file named file.txt.

Writes "Hello, World!" into file.txt.

Displays the contents of file.txt to the terminal.

* ls -l | grep ".txt  
  Lists all files and directories in long format and filters the list to show only items with .txt in their names.
* grep -r "pattern" /path/to/directory/  
  Recursively searches for the string "pattern" in all files within /path/to/directory/ and its subdirectories.
* cat file1.txt file2.txt | sort | uniq –d  
  Concatenates the contents of file1.txt and file2.txt.

Sorts the combined contents.

Displays only the lines that are repeated (duplicates).

* chmod 644 file.txt  
  Changes the permissions of file.txt to 644, which means:

Owner: read and write (6).

Group: read-only (4).

Others: read-only (4).

* cp -r source\_directory destination\_directory  
  Recursively copies the contents of source\_directory to destination\_directory. If destination\_directory doesn’t exist, it’s created.
* find /path/to/search -name "\*.txt"  
  Searches for all files with a .txt extension within /path/to/search and its subdirectories
* chmod u+x file.txt

Adds execute permission to the file file.txt for the owner (u stands for user).

* echo $PATH  
  Displays the current PATH environment variable, which lists the directories the shell searches to find executable files.

**Part B**

True or False Statements:

1. True: ls is used to list files and directories in a directory.
2. True: mv is used to move files and directories.
3. False: cd is used to change the current directory, not to copy files and directories.
4. True: pwd stands for "print working directory" and displays the current directory.
5. True: grep is used to search for patterns in files.
6. True: chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.
7. True: mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.
8. True: rm -rf file.txt deletes a file forcefully without confirmation.

## Identify the Incorrect Commands:

1. **chmodx** is incorrect. The correct command to change file permissions is chmod.

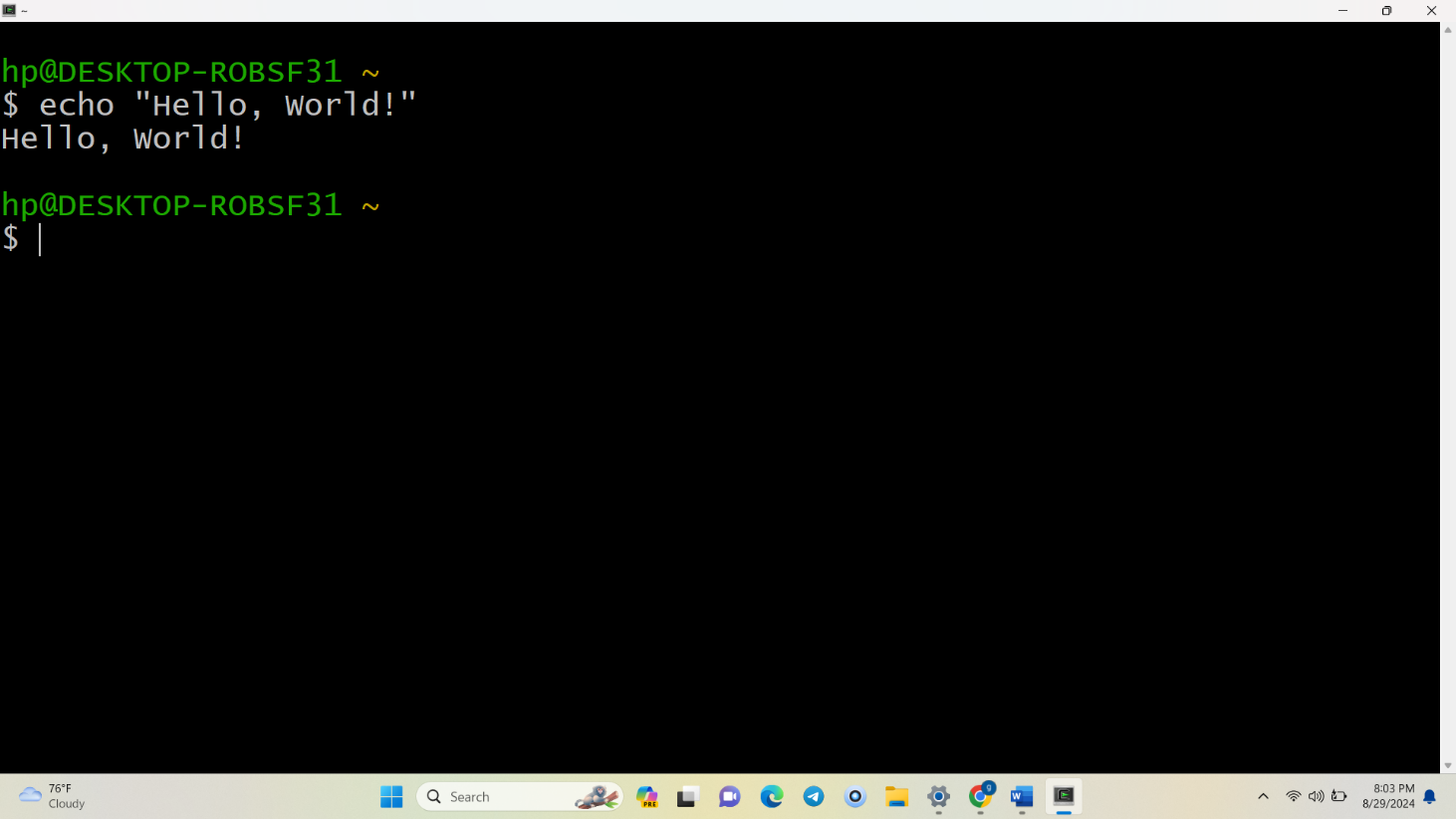
2. **cpy** is incorrect. The correct command to copy files and directories is cp.

3. **mkfile** is incorrect. The correct way to create a new file is using touch or echo "content" > filename.

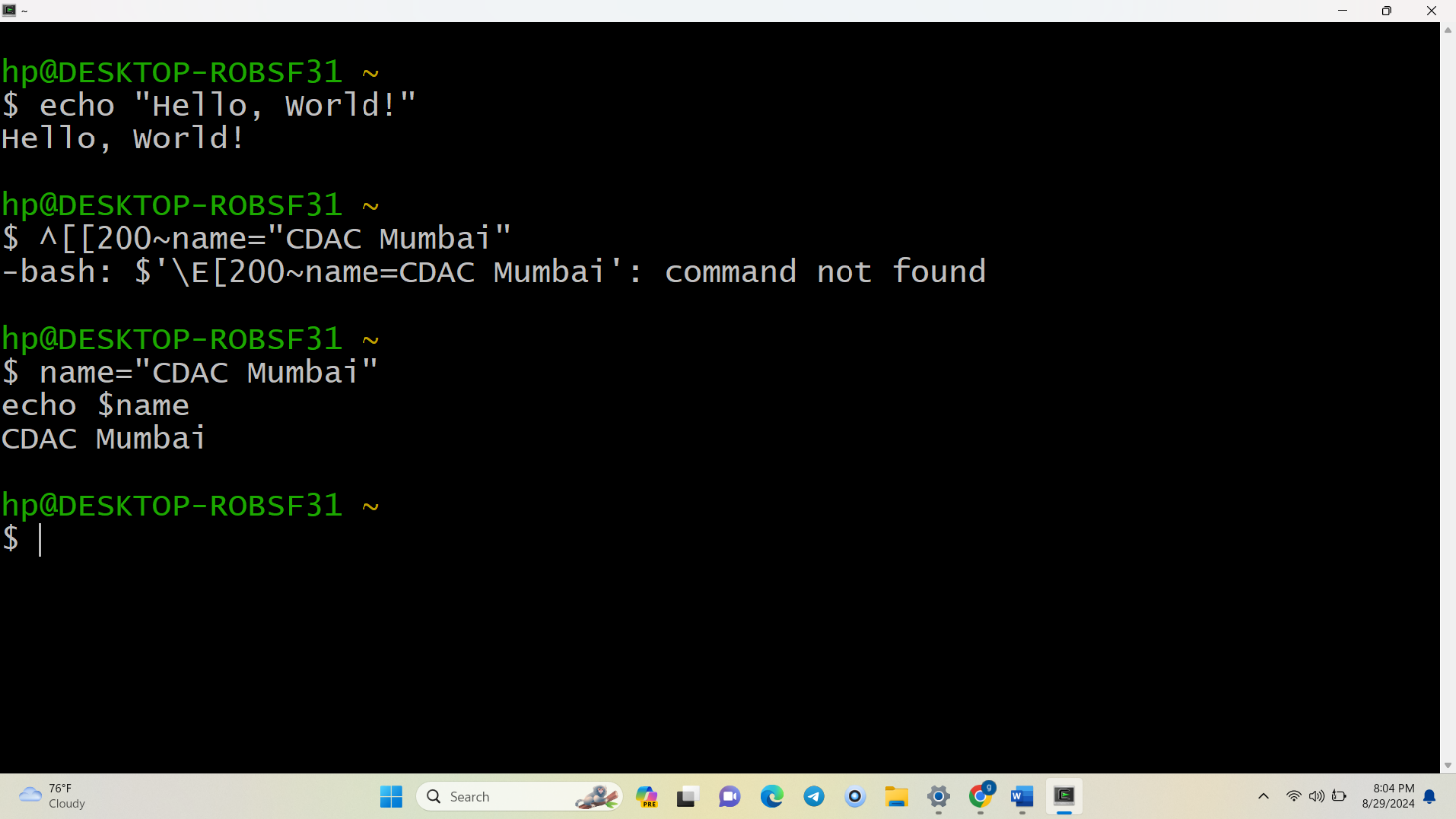
4. **catx** is incorrect. The correct command to concatenate files is cat.

5. **rn** is incorrect. The correct command to rename files is mv (or rename in some systems).

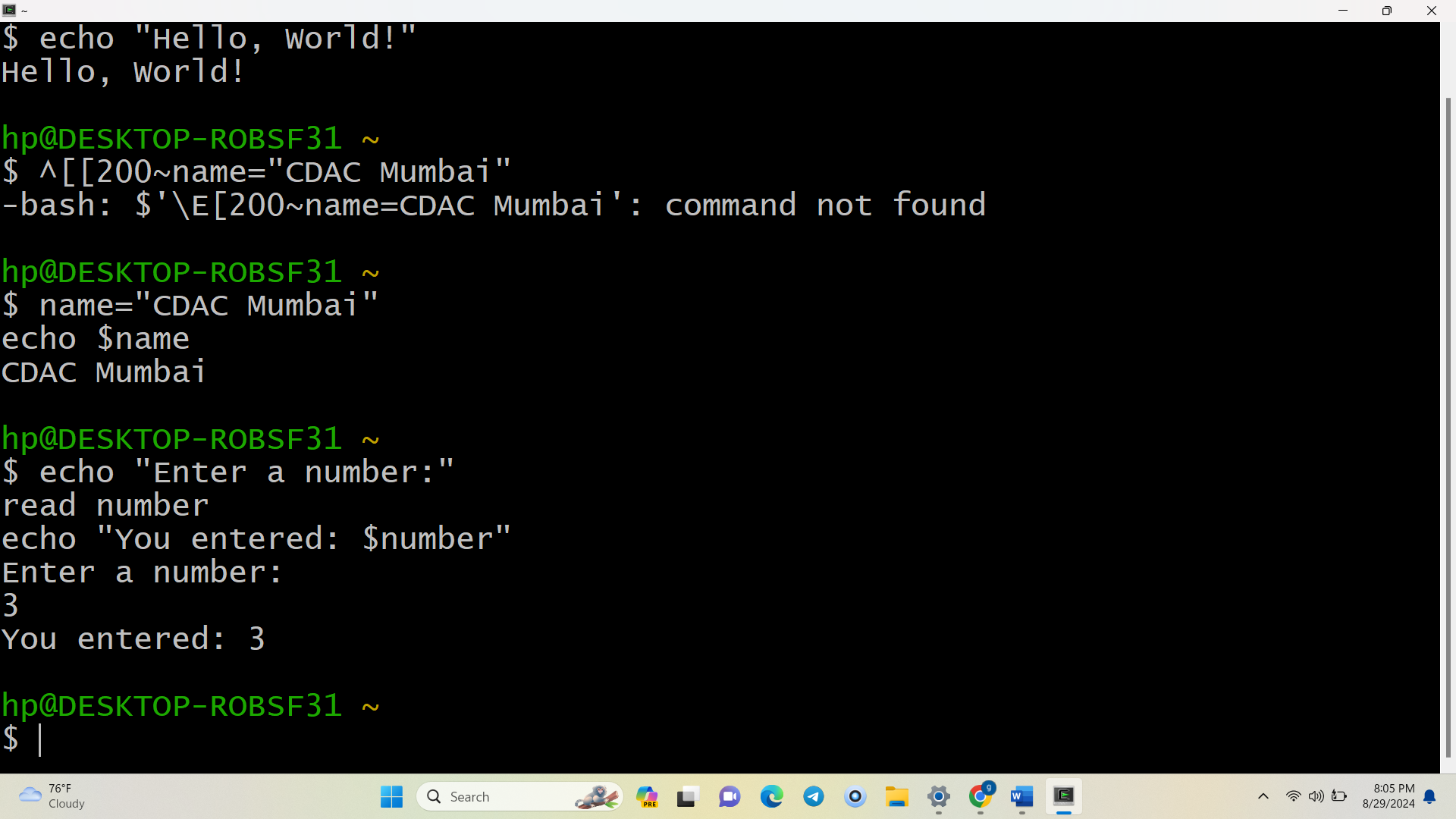
**Part C  
  
Question 1:** Write a shell script that prints "Hello, World!" to the terminal.



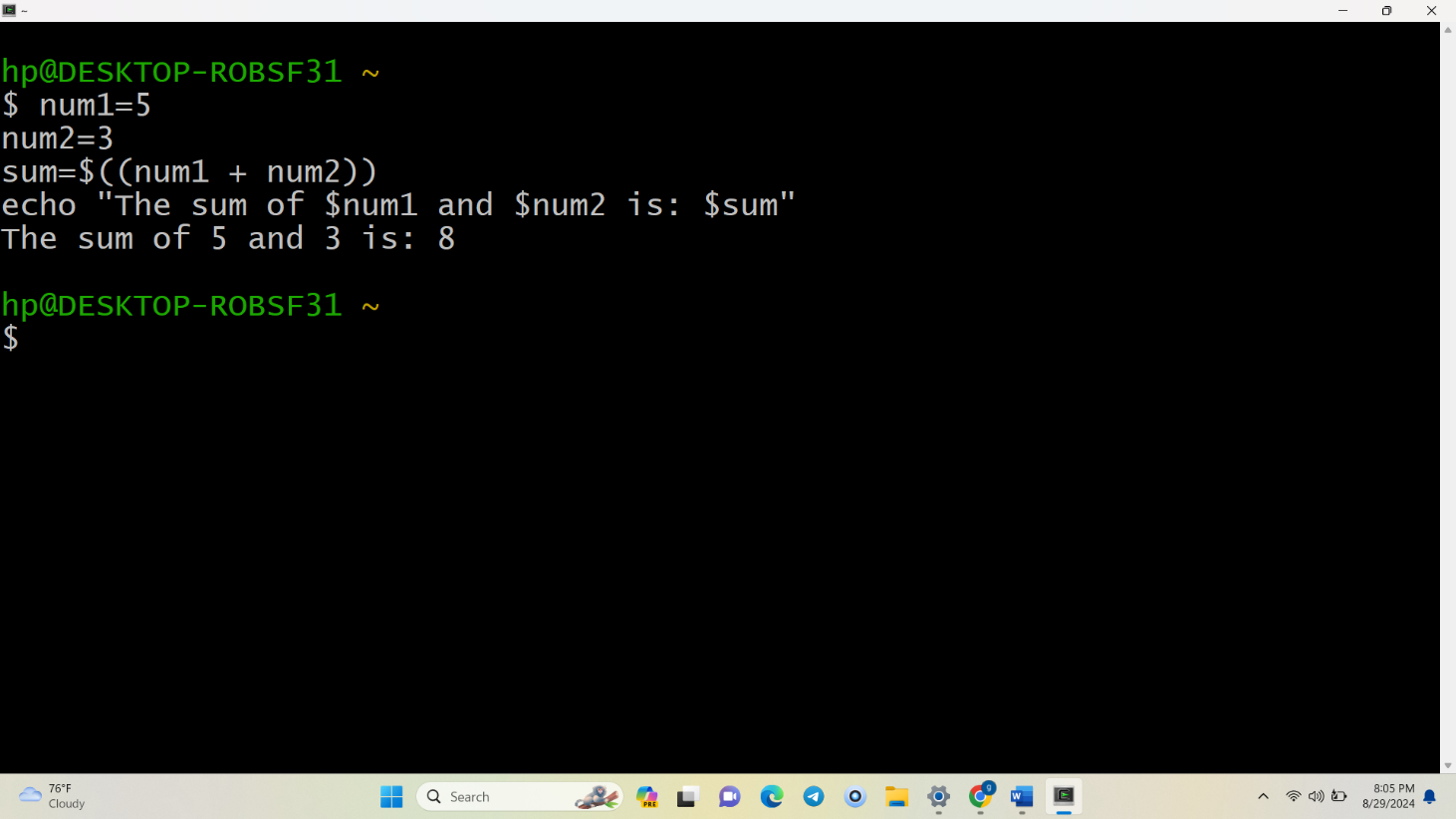
**Question 2:** Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.



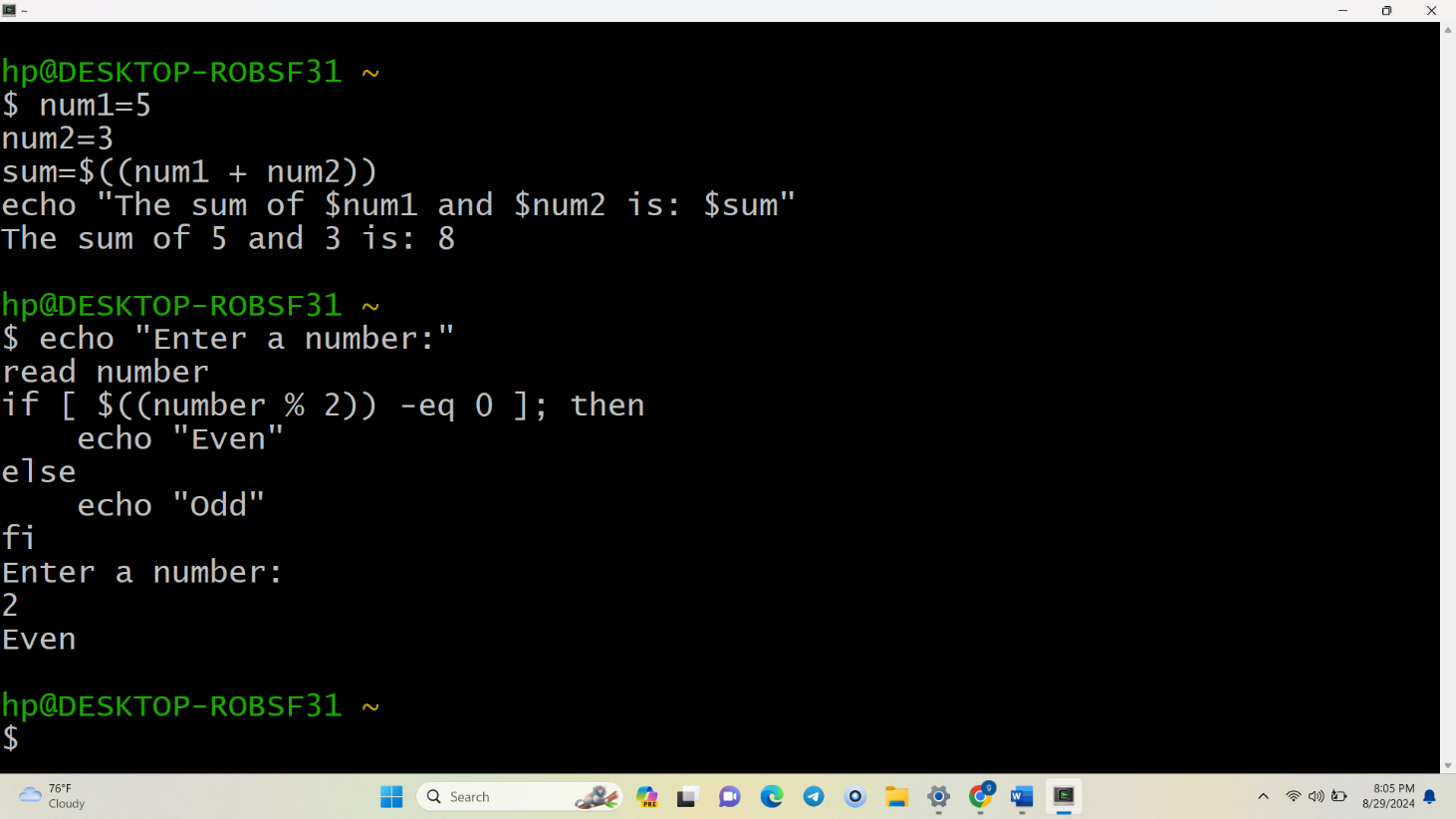
**Question 3:** Write a shell script that takes a number as input from the user and prints it.



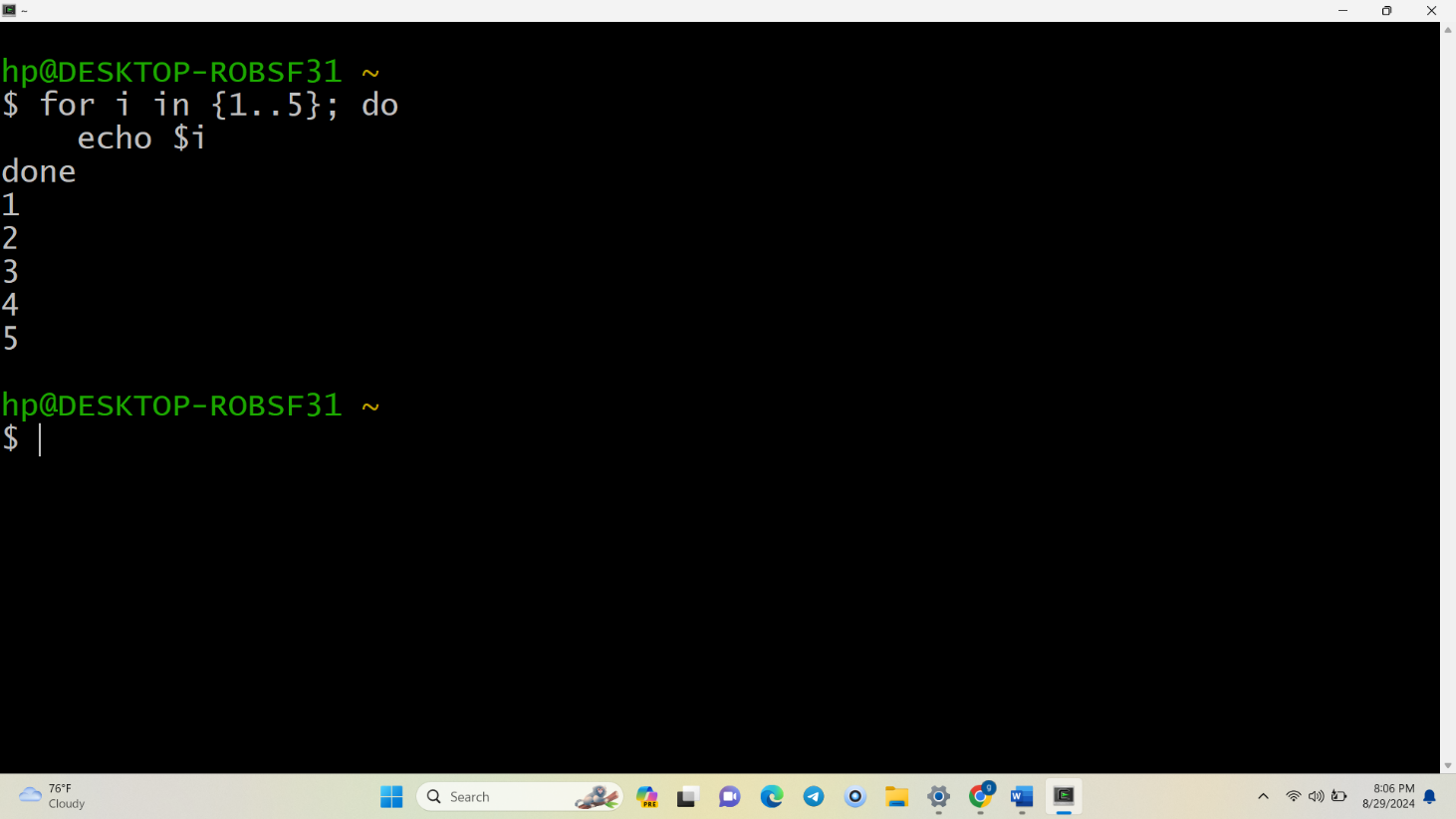
**Question 4:** Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.



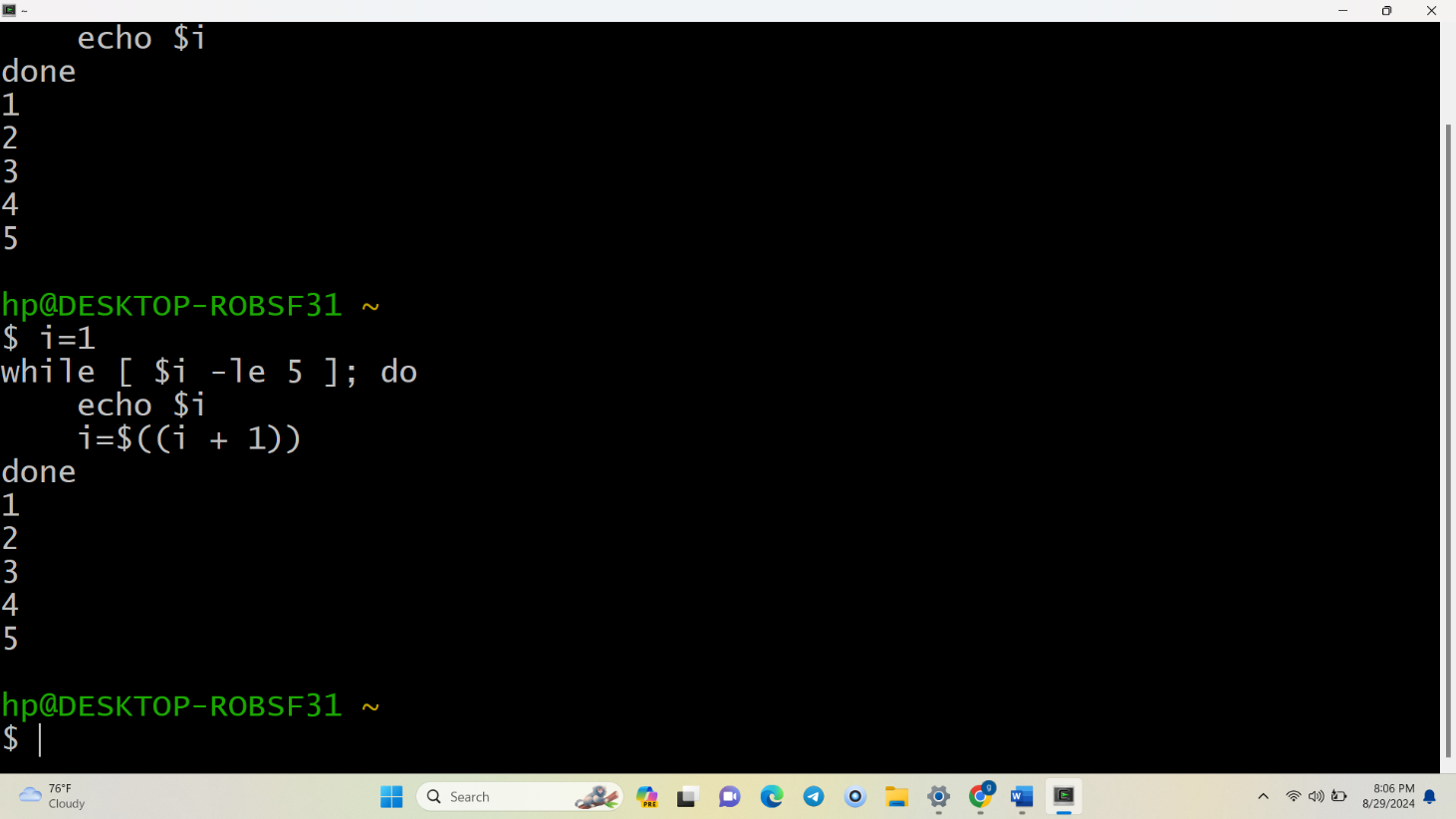
**Question 5:** Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".



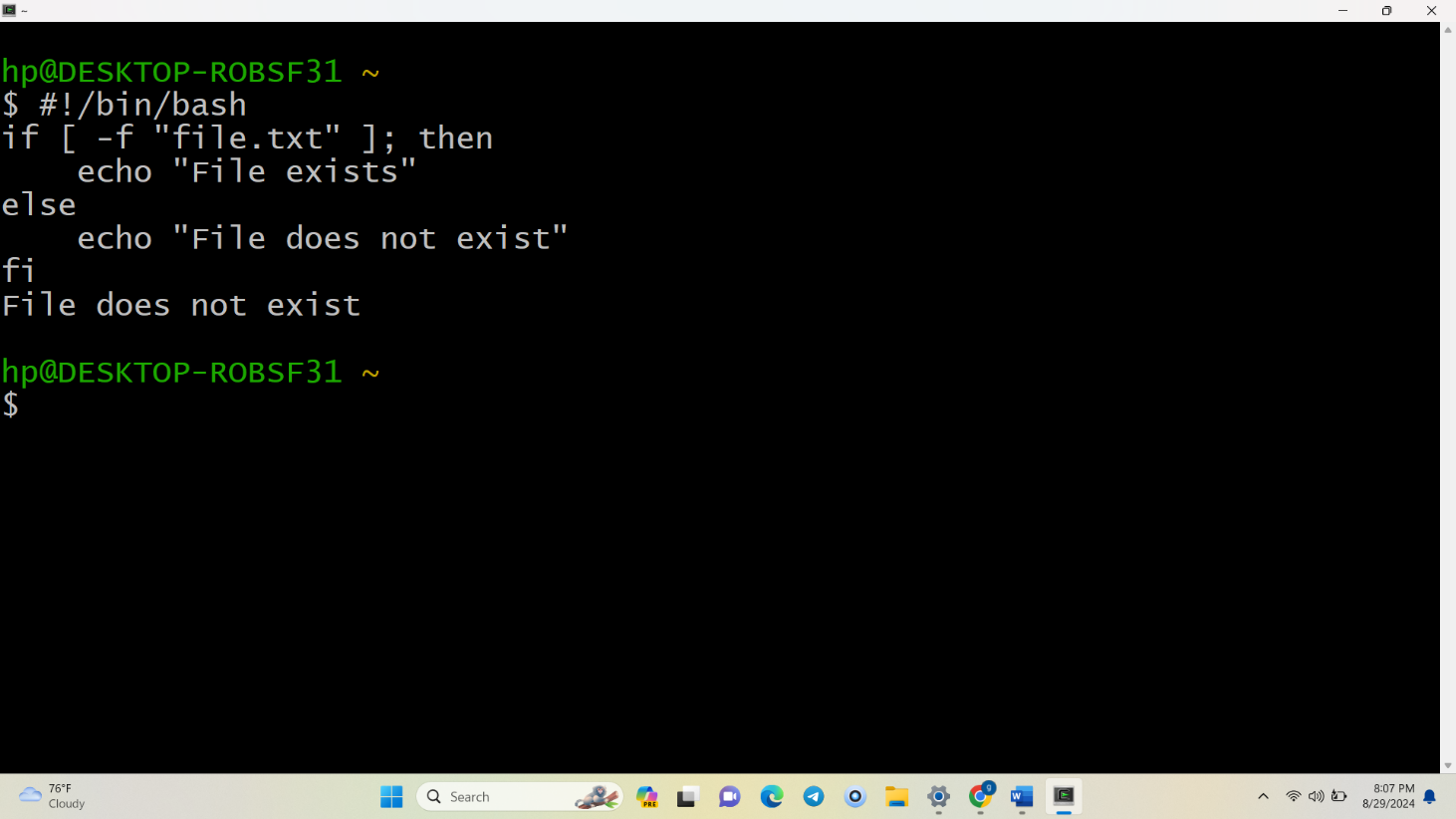
**Question 6:** Write a shell script that uses a for loop to print numbers from 1 to 5.



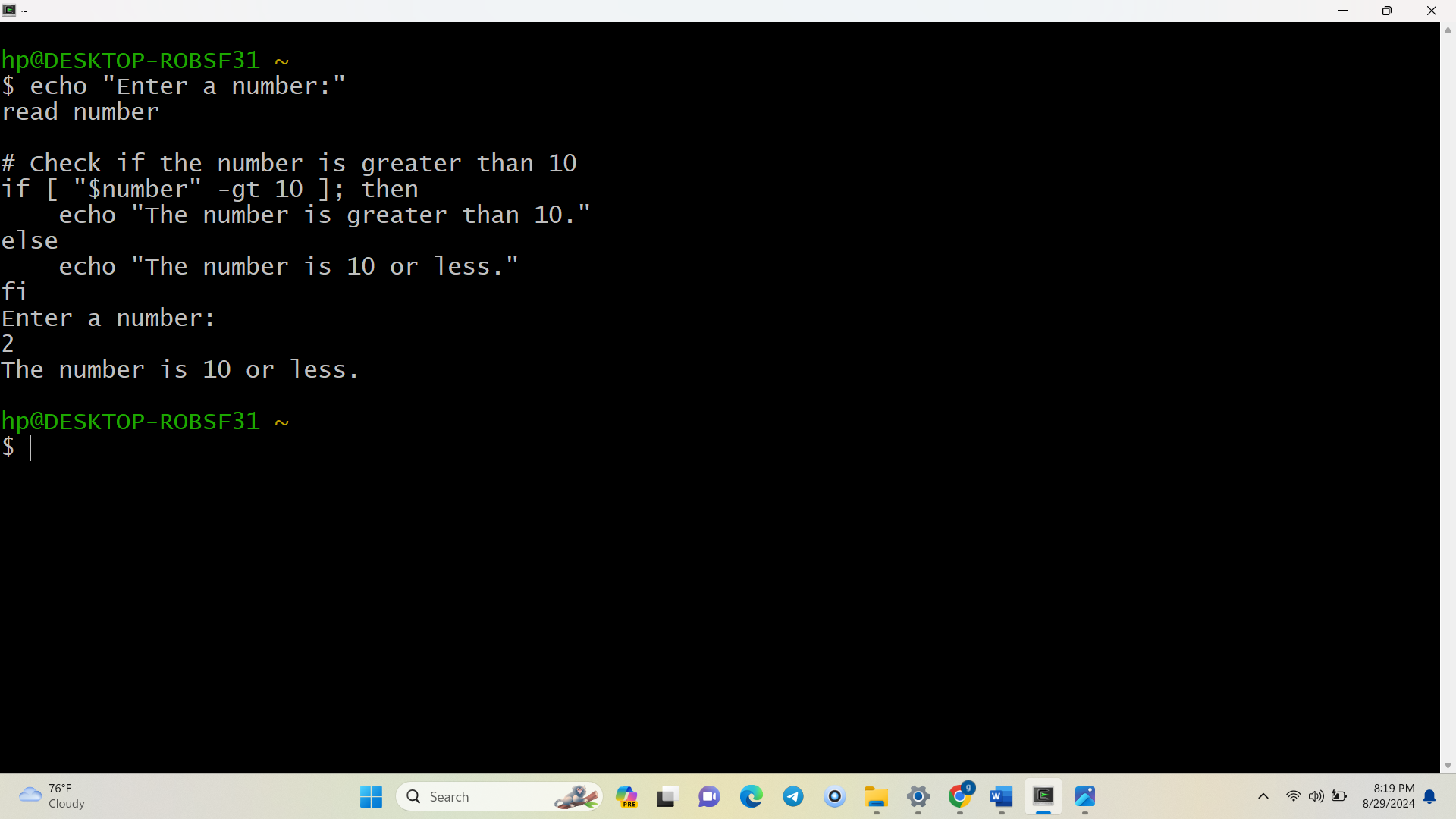
**Question 7:** Write a shell script that uses a while loop to print numbers from 1 to 5.



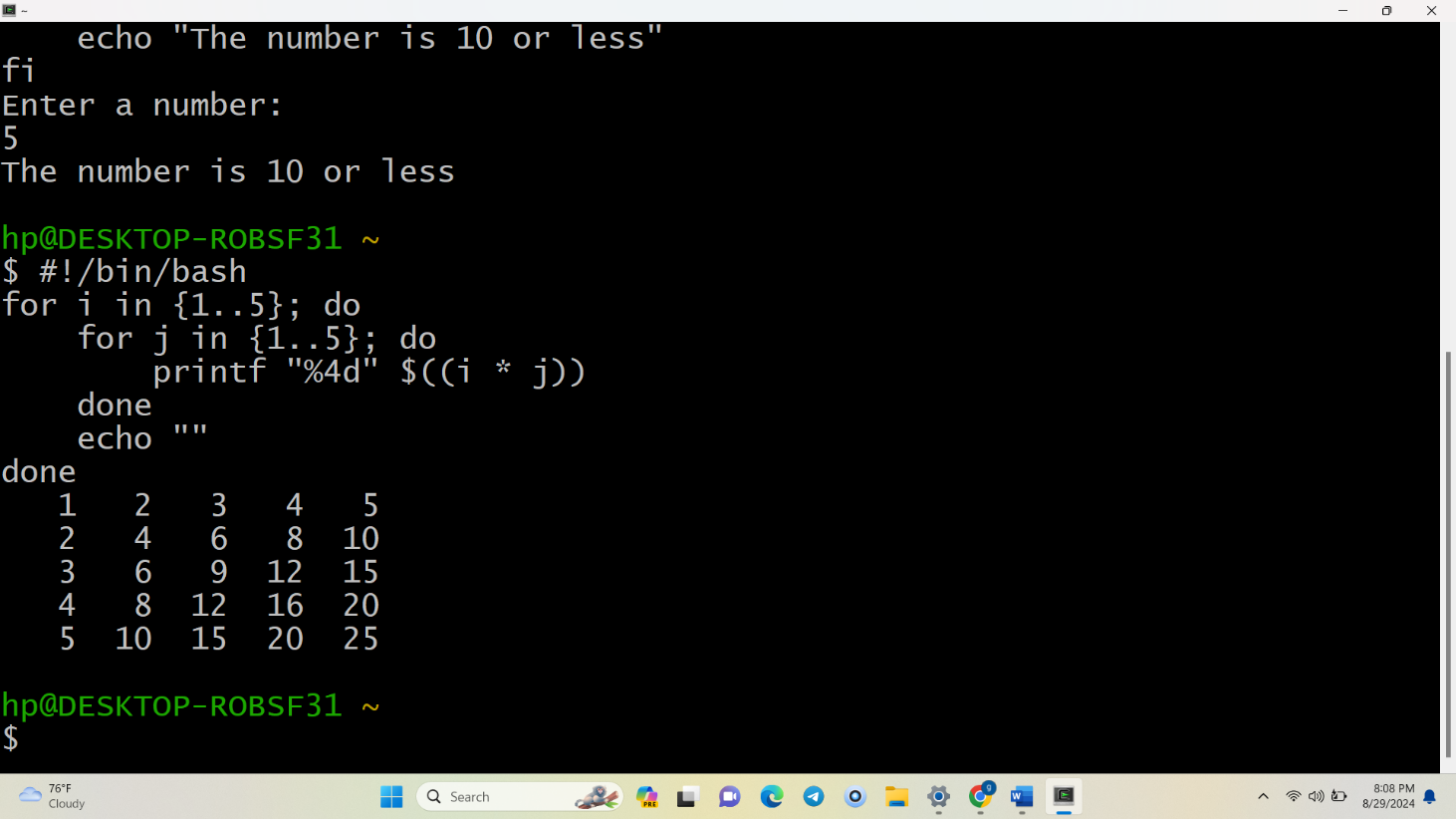
**Question 8:** Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



**Question 9:** Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



**Question 10:** Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.



**Question 11:** Write a shell script that uses a while loop to read numbers from the user until the user enters a negative number. For each positive number entered, print its square. Use the **break** statement to exit the loop when a negative number is entered.

