# PART A

**What will the following commands do?**

* **echo "Hello, World!"**

Ans: this will print Hello world to the console in shell

* **name="Productive"**

Ans: setting a value to variable name

* **touch file.txt**

Ans: it is used to create a file named file.txt

* **ls -a**

Ans: it is used to list all files in durectory

* **rm file.txt**

Ans: it is used to delete a file named file.txt

* **cp file1.txt file2.txt**

Ans: it is used to copy files. File1 and file2

* **mv file.txt /path/to/directory/**

Ans: it is used to move file from one directory to another

* **chmod 755 script.sh**

Ans: it is used to change permission of file of all

* **grep "pattern" file.txt**

Ans: it will print highlight the word pattern in file.txt

* **kill PID**

Ans: The kill command kills a single process at a time with the given process ID.

* **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**

Ans:

* **ls -l | grep ".txt"**

Ans: this command will list out all the .txt files

* **cat file1.txt file2.txt | sort | uniq**

Ans: this command will sort the unique content from file1 and file2

* **ls -l | grep "^d"**

Ans: this command will list out all the files starting with d

* **grep -r "pattern" /path/to/directory/**

Ans: this will print path of all the locations where word pattern is present

* **cat file1.txt file2.txt | sort | uniq –d**

Ans: this command will return unique lines in sorted order from file1 and file 2.

* **chmod 644 file.txt**

Ans: this command modifies the permission owner can read and write and all others only can read.

* **cp -r source\_directory destination\_directory**

Ans: this command will copy from one directory to another directory

* **find /path/to/search -name "\*.txt"**

Ans: this command will search for .txt files in given directory

* **chmod u+x file.txt**

Ans: this command gives owner a permission to execute file

* **echo $PATH**

Ans: this is used to print PATH variable in shell

# PART B

Identify True or False:

1. ls is used to list files and directories in a directory. **[True]**

2. mv is used to move files and directories. **[True]**

3. cd is used to copy files and directories. **[False]**

4. pwd stands for "print working directory" and displays the current directory. **[True]**

5. grep is used to search for patterns in files. **[True]**

6. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others. **[True]**

7. mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist. **[True]**

8. rm -rf file.txt deletes a file forcefully without confirmation. **[True]**

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

Ans: **chmod**

2. cpy is used to copy files and directories.

Ans: **cp**

3. mkfile is used to create a new file.

Ans: **touch**

4. catx is used to concatenate files.

Ans: **cat**

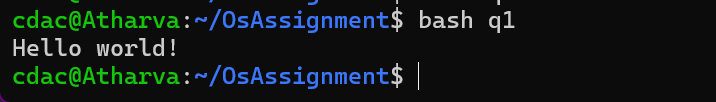
5. rn is used to rename files.

Ans: **mv**

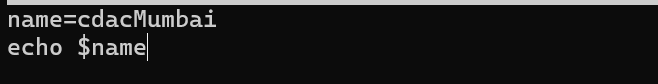
# PART C

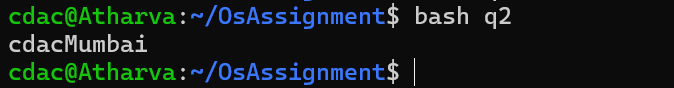
Question 1: Write a shell script that prints "Hello, World!" to the terminal.



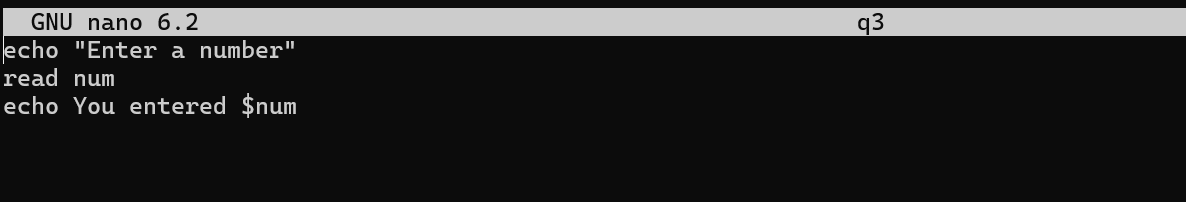


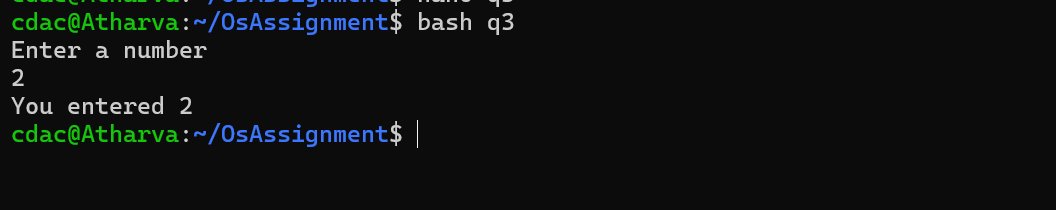
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.



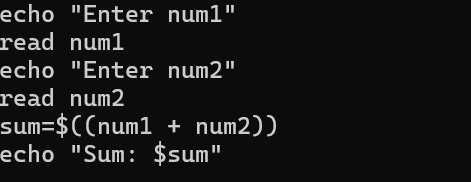


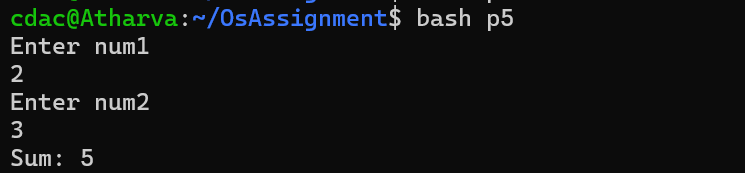
Question 3: Write a shell script that takes a number as input from the user and prints it.



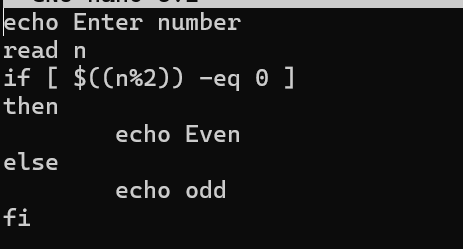


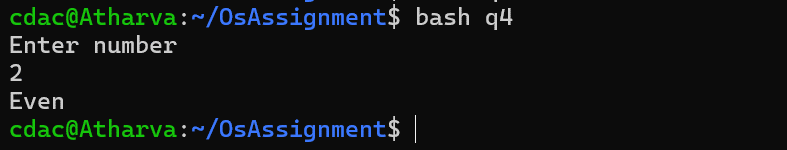
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.



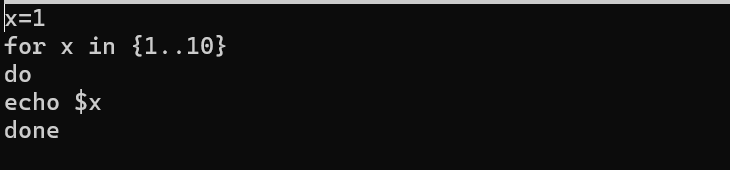


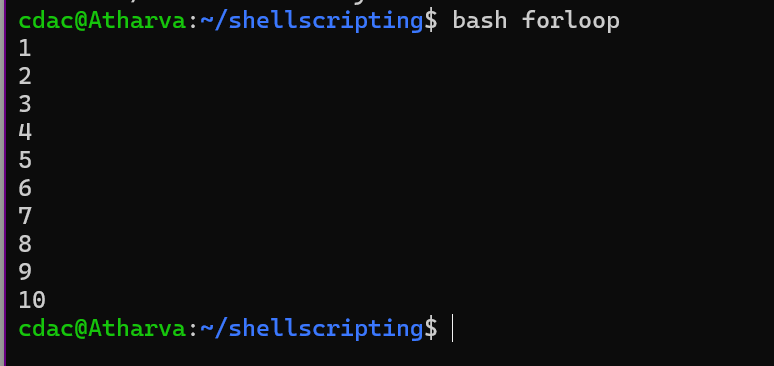
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd"



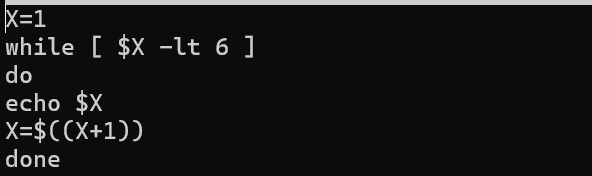


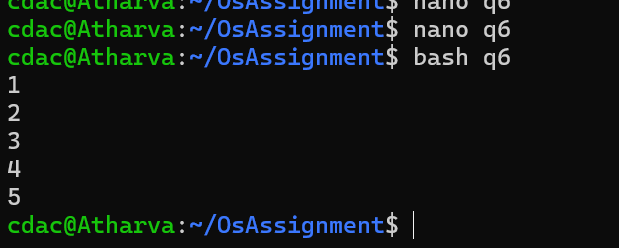
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.



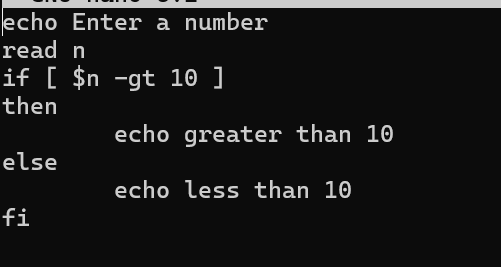


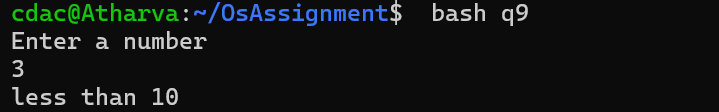
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



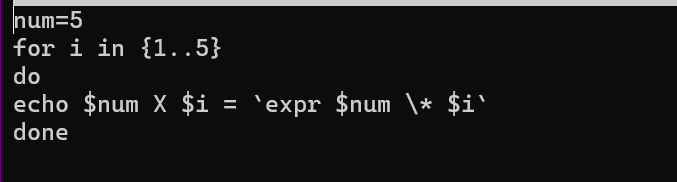


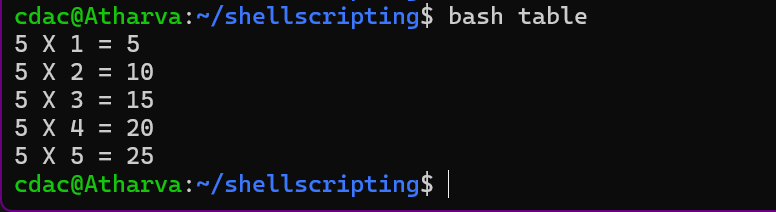
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



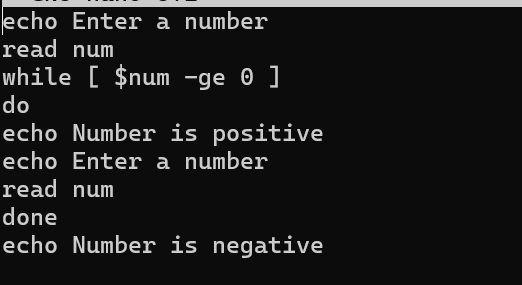


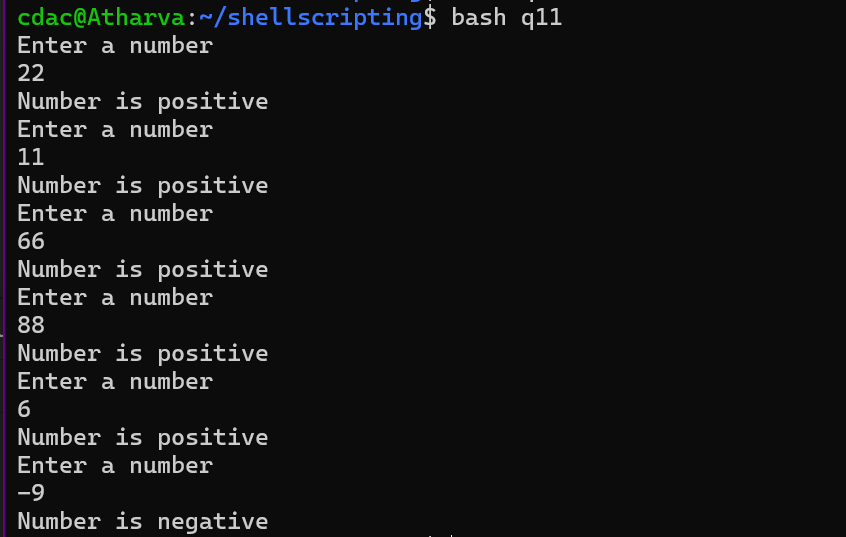
Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.





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.





# PART E

1. Consider the following processes with arrival times and burst times:

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PID | Arrival time | Burst Time | Waiting Time | Turn Around Time |
| P1 | 0 | 5 | 0 | 5 |
| P2 | 1 | 3 | 4 | 7 |
| P3 | 2 | 6 | 6 | 12 |

**Average waiting time = 3.33**

Gantt chart

|  |  |  |
| --- | --- | --- |
| P1 | P2 | P3 |
| 0 5 | 8 | 14 |

2. Consider the following processes with arrival times and burst times: Calculate the average turnaround time using Shortest Job First (SJF) scheduling.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PID | Arrival time | Burst Time | | Waiting Time | Turn Around Time |
| P1 | 0 | 3 | | 0 | 3 |
| P2 | 1 | 5 | | 7 | 12 |
| P3 | 2 | 1 | | 1 | 2 |
| P4 | 3 | | 4 | 1 | 5 |

**Average TAT= 5.5**

**Gantt chart**

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P3 | P4 | P2 |
| 0 3 | 4 | 8 | 13 |

1. Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority)

calculate the average waiting time using Priority Scheduling.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PID | Arrival time | Burst Time | priority | Waiting time |
| P1 | 0 | 6 | 3 | 13 |
| P2 | 1 | 4 | 1 | 0 |
| P3 | 2 | 7 | 4 | 5 |
| P4 | 3 | 2 | 2 | 2 |

**Average waiting time = 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P1** | **P2** | **P4** | **P3** | **P1** |
| **0 1** | **5** | **7** | **14** | **19** |

|  |
| --- |
|  |

: process complete

1. Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2 units: Calculate the average turnaround time using Round Robin scheduling.

|  |  |  |  |
| --- | --- | --- | --- |
| PID | Arrival Time | Burst Time | TAT |
| P1 | 0 | 4 | 10 |
| P2 | 1 | 5 | 13 |
| P3 | 2 | 2 | 4 |
| P4 | 3 | 3 | 11 |

**Average TAT= 9.5**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **P1** | **P2** | **P3** | **P4** | **P1** | **P2** | **P4** | **P2** |
| **0 2** | **4** | **6** | **8** | **10** | **12** | **14** | **16** |

|  |
| --- |
|  |

: process complete

1. Consider a program that uses the fork() system call to create a child process. Initially, the parent process has a variable x with a value of 5. After forking, both the parent and child processes increment the value of x by 1. What will be the final values of x in the parent and child processes after the fork() call?

Ans: **11**