**Concepts of Operating System**

**Assignment 2**

**Part A**

**What will the following commands do?**

1. **echo "Hello, World!"**

A command used to display a line of text or a string on the screen.

1. **name="Productive"**

assigning the value, you can use the variable name

1. **touch file.txt**

* A command that creates a new empty file or updates the of an existing file**.**
* file doesn't exist: The command will create an empty fille
* file already exists: The command will update the file's last modified time to the current time, but the contents of the file will remain unchanged.

1. **ls -a**

The command will display all files and directories in the current directory, including hidden ones.

1. **rm file.txt**

The command used to remove files or directories**.**

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

* cp: The command used to copy files or directories.
* file1.txt: The source file that you want to copy.
* file2.txt: The destination file where the contents of file1.txt will be copied

1. **mv file.txt /path/to/directory/**
   * mv : is the command for moving files or directories.
   * file.txt: is the name of the file you want to move.
   * /path/to/directory/ : is the destination directory where you want to move the file.

**7.chmod 755 script.sh**

* chmod : is the command used to change file permissions.
* 755 : is the permission setting.
* script.sh is the file whose permissions you're modifying.

**What 755 Means**:

* The first digit (7) gives the owner (user) read (r), write (w), and execute (x) permissions.
* The second digit (5) gives the group read (r) and execute (x) permissions, but no write permission.
* The third digit (5) gives others (everyone else) read (r) and execute (x) permissions, but no write permission.

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

* grep is the command used for searching text using patterns.
* "pattern" is the string or regular expression you're looking for within the file.
* file.txt is the file in which grep will search for the specified pattern.

**9. kill PID**

to terminate a process

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

* mkdir mydir : Creates a new directory named mydir.
* cd mydir : Changes the current directory to mydir.
* touch file.txt : Creates an empty file named file.txt in the mydir directory.
* echo "Hello, World!" > file.txt: Writes the text "Hello, World!" into file.txt, overwriting any existing content in the file.
* cat file.txt: Displays the contents of file.txt

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

* Lists all files and directories in the current directory with detailed information, including permissions, number of links, owner, group, size, and timestamp.
* | **(pipe)**: Takes the output of the ls -l command and passes it as input to the grep command.
* grep ".txt": Searches through the output from ls -l and display ls -l: ys only the lines that contain .txt, which corresponds to files with a .txt extension.

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

* cat file1.txt file2.txt:Concatenates the contents of file1.txt and file2.txt and outputs them together.
* | (pipe):Takes the output from the cat command and passes it as input to the sort command.
* sort:Sorts the lines of the input data in ascending order. This is necessary because uniq only removes adjacent duplicate lines.
* | (pipe):Takes the sorted output from the sort command and passes it as input to the uniq command.
* Uniq :Filters out adjacent duplicate lines, displaying only unique lines from the sorted output.

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

* ls -l:Lists all files and directories in the current directory with detailed information, including permissions, number of links, owner, group, size, and timestamp.
* | (pipe):Takes the output of ls -l and passes it as input to the grep command.
* grep "^d":Searches through the output for lines that start with d. In the output of ls -l, directories are indicated by a d at the beginning of the line (in the file type field of the permissions).

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

* grep:The command used for searching text using patterns.
* -r:This option stands for "recursive" and tells grep to search through all files and subdirectories within the specified directory.
* "pattern": The text or pattern you want to search for.
* /path/to/directory/:The directory where grep should start the search.

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

* cat file1.txt file2.txt:
* Concatenates the contents of file1.txt and file2.txt, outputting them together.
* | (pipe):
* Passes the concatenated output from cat to the sort command.
* sort:
* Sorts the lines of the input data in ascending order. Sorting is necessary for uniq to properly identify duplicate lines.
* | (pipe):
* Passes the sorted output to the uniq command.
* uniq -d:
* Filters out and displays only the lines that are duplicated (i.e., lines that appear more than once) in the sorted data.

1. **chmod 644 file.txt**

* chmod: The command used to change file permissions.
* 644: The permission setting you want to apply.
* file.txt: The file to which you are applying the permissions.
* What 644 Means:
* 6: Sets the owner's permissions to read (r) and write (w), but not execute (x). This is represented as rw-.
* 4: Sets the group’s permissions to read (r) only, with no write or execute permissions. This is represented as r--.
* 4: Sets others’ permissions to read (r) only, with no write or execute permissions. This is represented as r--.

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

* **cp**: The command used to copy files and directories.
* **-r**: Stands for "recursive" and tells cp to copy directories and their contents recursively. This means that all files, subdirectories, and their contents within the source\_directory will be copied to the destination\_directory.
* **source\_directory**: The directory you want to copy.
* **destination\_directory**: The directory where you want to copy source\_directory. If destination\_directory does not exist, it will be created.

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

* **find**: The command used to search for files and directories in a directory hierarchy.
* **/path/to/search**: The directory where you want to start the search. This can be a full path or a relative path.
*  **-name "\*.txt"**: Specifies the search criterion. -name tells find to look for files matching the given pattern, which in this case is "\*.txt". This pattern matches all files with a .txt extension.

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

* **chmod**: The command used to change file permissions.
* **u+x**: The permission setting to be applied.
  + **u** stands for "user" (the file's owner).
  + **+x** adds execute (x) permissions.
* **file.txt**: The file to which you are applying the permission changes.

1. **echo $PATH**

* **echo**: Prints the value of a variable or text to the terminal.
* **$PATH**: Refers to the PATH environment variable, which defines the directories the shell searches for executable files.

**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**

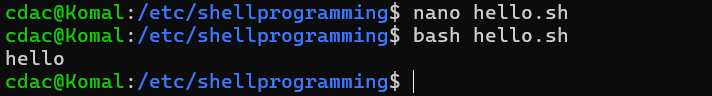
**8. rm -rf file.txt deletes a file forcefully without confirmation. => True**

**Identify the Incorrect Commands:**

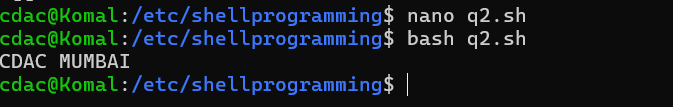
* **chmodx is used to change file permissions. chmod**
* **cpy is used to copy files and directories. cp**
* **mkfile is used to create a new file. touch**
* **catx is used to concatenate files. cat**
* **rn is used to rename files. mv**

**Part C**

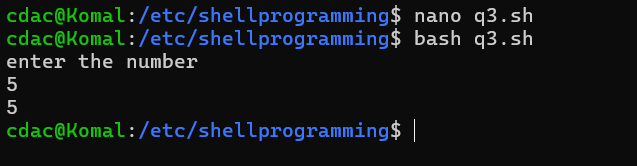
**Question 1: Write a shell script that prints "Hello, World!" to the terminal.**



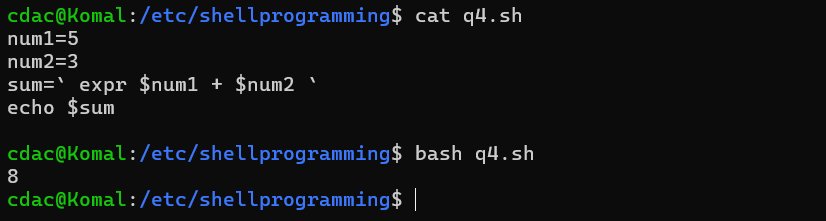
**Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.**

****

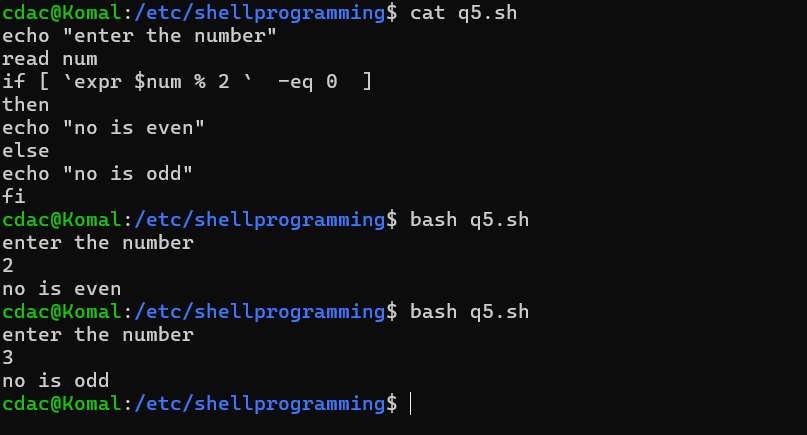
**Question 3: Write a shell script that takes a number as input from the user and prints it.**

****

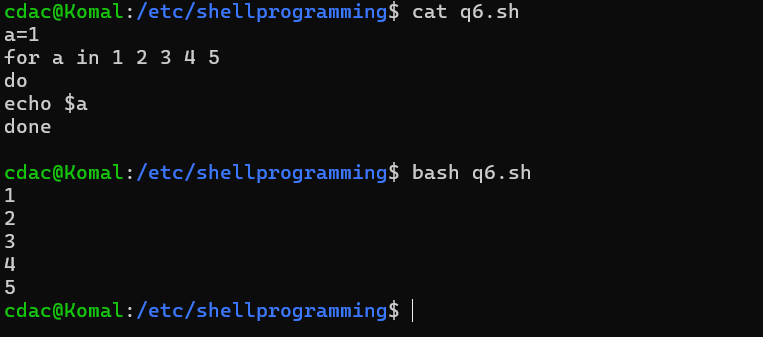
**Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.**

****

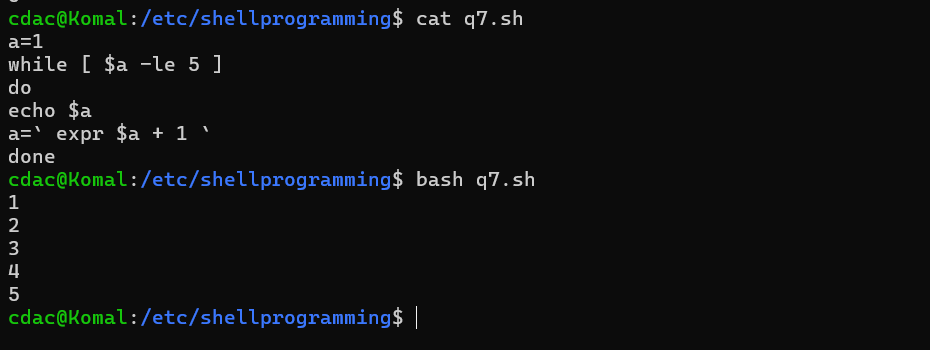
**Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".**

****

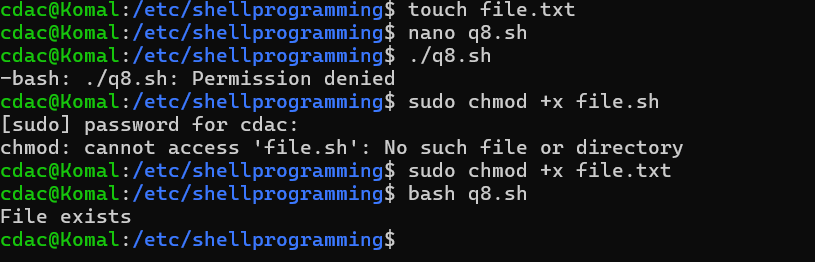
**Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.**

****

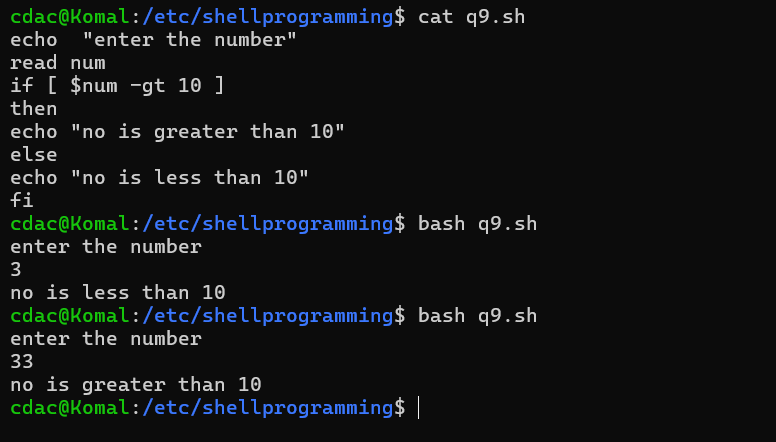
**Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.**

****

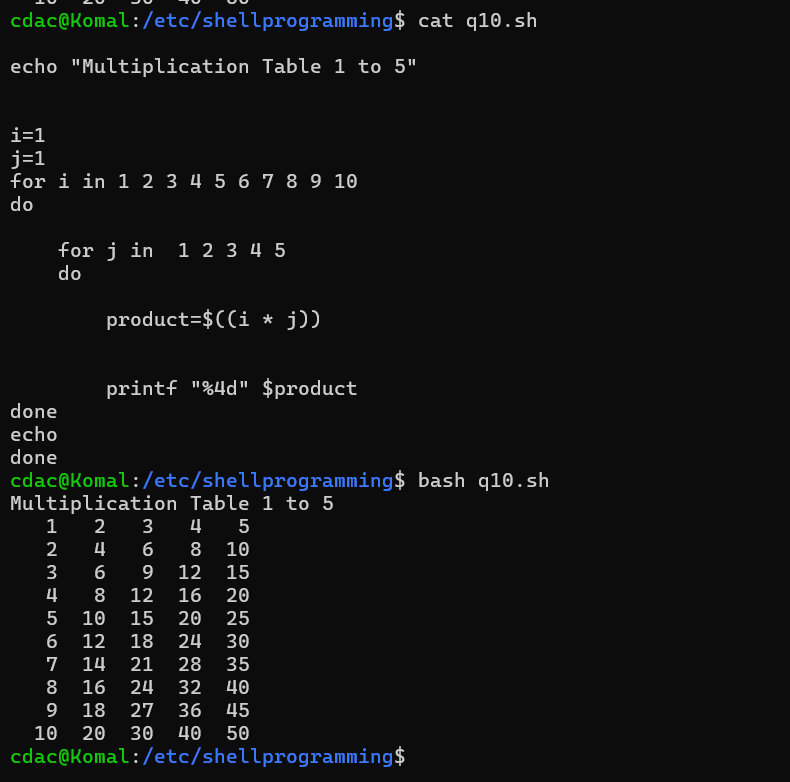
**Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".**

****

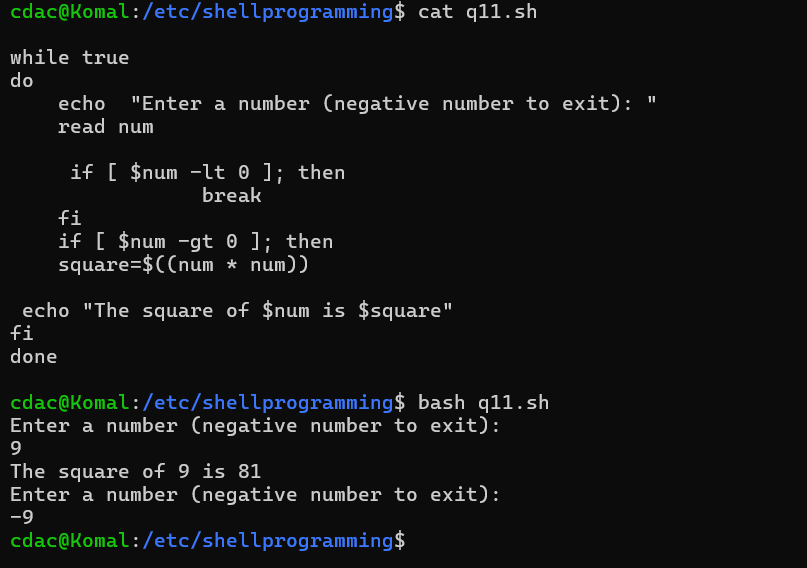
**Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.**

****

**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.**

****