**ASSIGNMENT 2**

**PART A**

What will the following command do?

* 1. **echo “Hello World”**
  2. The above command is used to display message on the output scree.
  3. **name=”Productive”**
  4. Here value “productive ” is assigned to the shell variable name.
  5. **touch file.txt**
  6. touch command will create an empty file.
  7. The above command is used to create file file.txt
  8. **ls –a**
  9. ls command lists the contents of a current directory.
  10. -a option return all the hidden files and directories.
  11. **rm file.txt**
  12. rm command is used to delete a file or directory (-r option). In the above example, rm command deletes the file named file.txt.
  13. **cp file1.txt file2.txt**
  14. cp command is used to copy files and directories.
  15. The cp command copies the content of file1.txt to file2.txt
  16. **mv file.txt /path/to/directory/**
  17. mv command is used rename or move a file. In the above example, mv command moves the file (file.txt) into the specified directory (/path/to/directory/).
  18. **chmod 755 script.sh**
  19. chmod command is used to change the permission of particular file or directory.The above command gives read,write,execute permission to the owner and read ,execute permission to the other and group.
  20. **grep "pattern" file.txt**
  21. grep command is used to search for specific patterns or regular expressions in text files & display the matching lines.
  22. The above command will return pattern matching string in file.txt
  23. **kill PID**
  24. This command will terminate the process whose PID is mentioned in the command. Since the above command doesn’t contain any process id, above command will result in an error.
  25. **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**
  26. && (logical AND) operator is used here which enables the user to run multiple commands in single command.
  27. The above command gives output of first command as input to the second command.here mkdir is used to create mydir directory,then cd will change the directory mydir,next touch command is used to create file file file.txt,next echo command is used to display output within the file file.txt and cat command is used to display the content of file on output screen.
  28. **ls -l | grep ".txt"**
  29. The above command uses piping to combine the output of both ls and grep command. ls -l is used to display the contents of current directory with details and grep “.txt” command is used to display all the files conating .txt pattern in their name.
  30. **cat file1.txt file2.txt | sort | uniq**
  31. The above command uses piping to combine the output of cat sort and uniq commands. First command i.e. cat command is used to display the contents of file1.txt followed by contents of file2.txt. sort command is used to perform alphanumeric sort on the result of cat command. Contents of file1.txt and file2.txt are sorted separately in the result. Finally, uniq command is use to display only distinct lines in the result.
  32. **ls -l | grep "^d"**
  33. ls command lists the files and directories in long format. grep "^d" command filters the output to show only lines that start with "d" which in the ls -l output indicates directories.
  34. **grep -r "pattern" /path/to/directory/**
  35. Here grep command is used to recursively search for given pattern “pattern” in the directory /path/to/directory, provided that such directory exists in first place. The output will display the lines containing the “pattern” pattern in it.
  36. **cat file1.txt file2.txt | sort | uniq –d**
  37. cat command displays the content of file1.txt followed by file2.txt. sort command is used to perform alphanumeric sort on the result of cat command. Contents of file1.txt and file2.txt are sorted separately in the result.
  38. uniq -d command is used to display only duplicate lines in the previous output.
  39. **chmod 644 file.txt**
  40. The above command assigns read and write permissions to owner of the file file.txt and read permission to group users and other users respectively.
  41. **cp -r source\_directory destination\_directory**
  42. The above command is used to copy the source\_directory to destination directory. This is done by using -r option so that all files in source\_directory are copied recursively.
  43. **find /path/to/search -name "\*.txt"**
  44. find command is used for searching the files and directories. Given command searches /path/to/search directory and its subdirectories for any file ending with .txt pattern.
  45. **chmod u+x file.txt**
  46. This command is used to grant execute permissions for file.txt file to the user(owner) of the file.
  47. **echo $PATH**
  48. This command displays the value of system environment variable that stores directories where executable programs are located.

**PART B**

**Identify True or False**

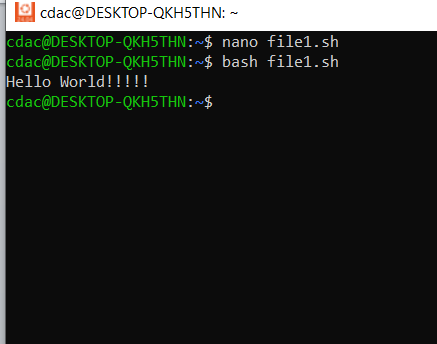
* ls is used to list files and directories in a directory. – **True**
* mv is used to move files and directories. – **True**
* cd is used to copy files and directories. – **False**, it is used to change the directory.
* pwd stands for "print working directory" and displays the current directory. – **True**
* grep is used to search for patterns in files. – **True**
* chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others. – **True**
* mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist. – **True**
* rm -rf file.txt deletes a file forcefully without confirmation. – **False**, -r (recursive option) is used for deleting directories, not files.

**Identify the Incorrect Command**

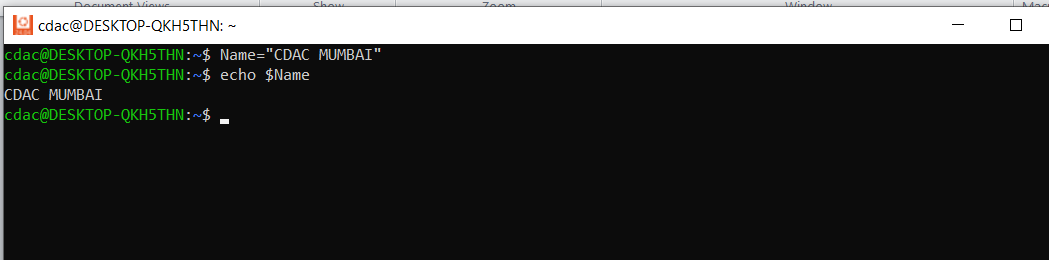
* 1. 1)chmodx is used to change file permissions. **chmod command is used to change file permissions.**
  2. 2)cpy is used to copy files and directories. **cp command is used to copy files and directories.**
  3. 3)mkfile is used to create a new file. **touch command is used to** **create a new file**. **mkdir command is used to create a new directory.**
  4. 4)catx is used to concatenate files. **cat command is used to concatenate files.**
  5. 5)rn is used to rename files. **mv command is used to rename files when 2 files names are passed as arguments.**

**PART C**

Q1. **Write a shell script that prints "Hello, World!" to the terminal.**



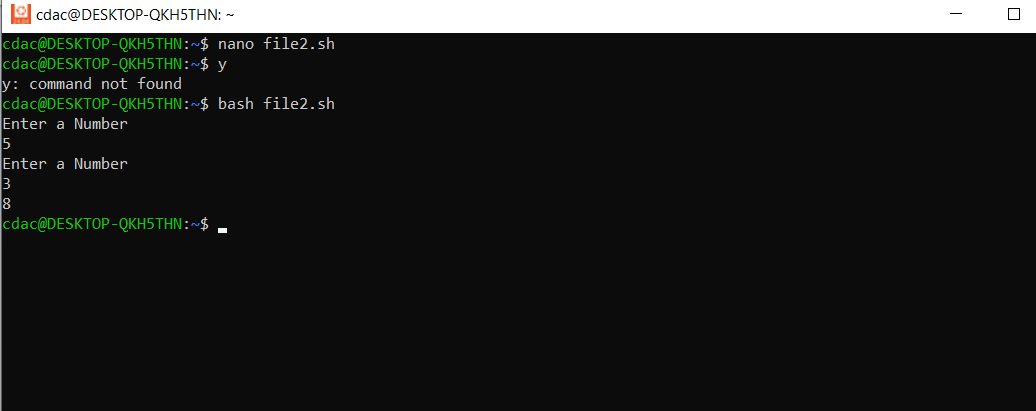
Q2. **Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.**



Q3. **Write a shell script that takes a number as input from the user and prints it.**



Q4. **Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.**

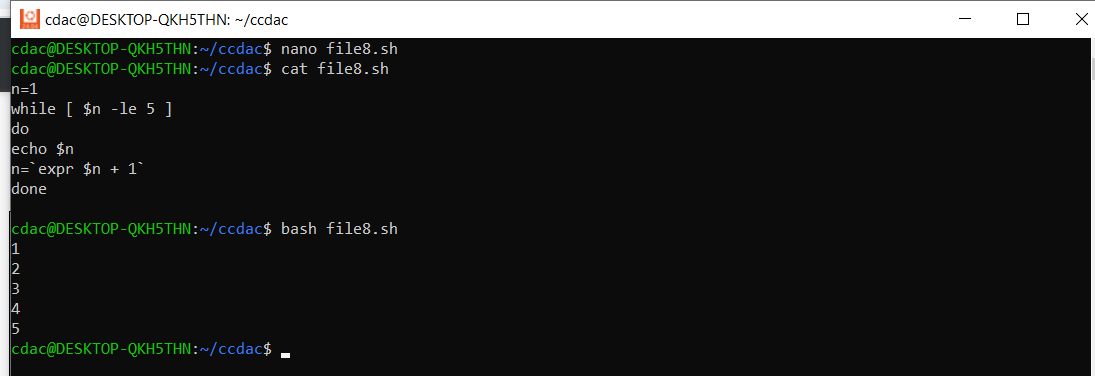


Q5. **Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".**

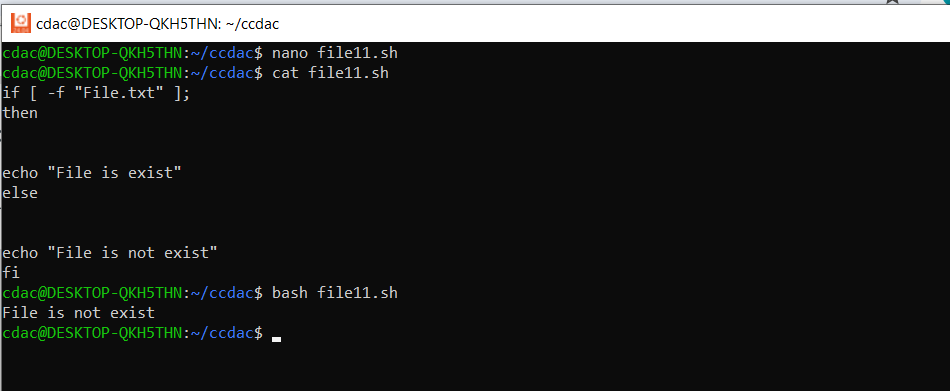


Q6. **Write a shell script that uses a for loop to print numbers from 1 to 5.**

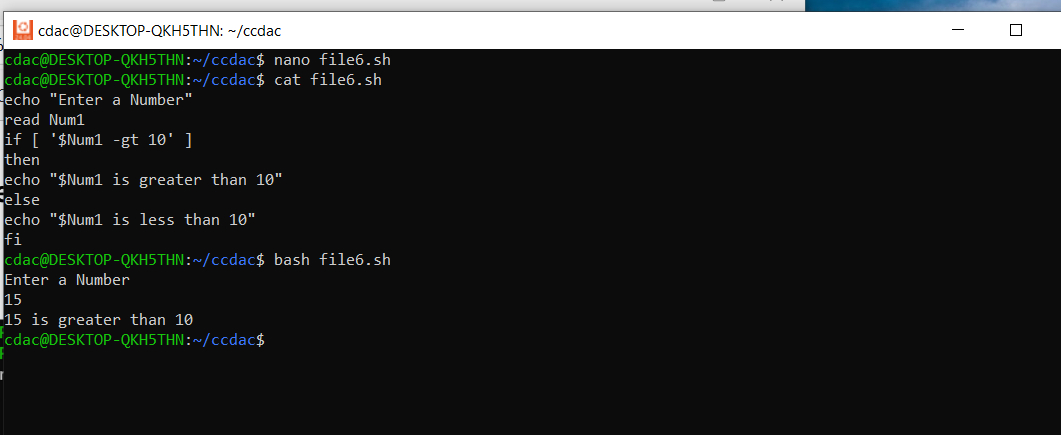


Q7. **Write a shell script that uses a while loop to print numbers from 1 to 5. **

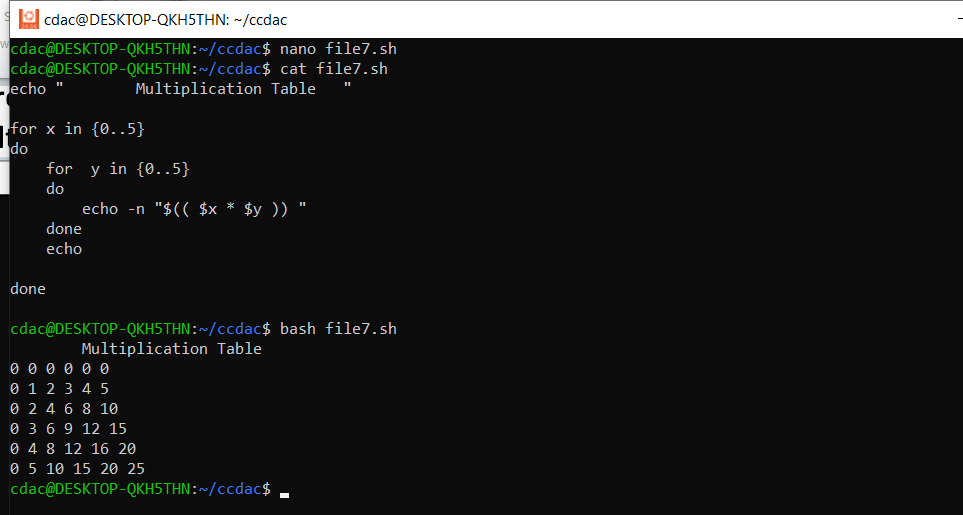
**Q8. Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".**



Q9. **Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.**



Q10. **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.**



Q11. **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.**

**🡺ANS:**

While [ true ]

do

echo “Enter a Number”

if[ $a –lt 0 ]

then

break

fi

done

echo “Program Termination”

**OUTPUT:**

Enter a number

10

Enter a number

2

Enter a number -4

Program Terminated

* 1. Q5. When the fork() system call is used, it creates a child process that has its own copy of the parent's memory.
  2. Before forking, the parent has a variable x = 5. After the fork, both the parent and child have separate copies of x, still equal to 5.
  3. Each process then increments x by 1, so both the parent and child have x = 6, but in their own separate memory.
  4. In parent process, x=6. In child process, x=6