**CDAC MUMBAI**

**Concepts of Operating System**

**Assignment No: 2**

**Part A**

**Name**: Akash Bhadane

**What will the following commands do?**

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

* Print the text Hello World! on the terminal.

1. **name="Productive"**

* Create a one Variable name and assign the value Production.

1. **touch file.txt**

* Create a new empty txt file named file.txt

1. **ls -a**

* list all files in the directory, including all hidden files.

1. **rm file.txt**

* Delete the file name file.txt.

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

* Copy the content of file1.txt into a new file file2.txt.

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

* Move the file file.txt into the specified directory.

1. **chmod 755 script.sh**

* change file permission so that:
* The owner can read, write, and execute.
* The group can read and execute.
* Others can read and execute

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

* Search for the word pattern inside file.txt prints the match line.

1. **kill PID**

* Terminates the process with the given process PID.

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

- Create a folder mydir.

- Change directory to mydir.

- Create a file named file.txt.

- Write Hello, World! Into the file.

- Displays the content of the file on the terminal.

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

* Shows file in long format but only displays those ending with .txt.

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

* Combines the content of two files, sorts the lines, and removes any duplicates.

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

* lists only directories or starting with d.

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

* Search the pattern word in all files under the given directory, including subfolder

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

* show only the lines that appear in both files or duplicates.

1. **chmod 644 file.txt**

* Set the file permissions using a numeric code 644.
* The owner can read and write.
* The group and others can only read.

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

* copy an entire folder to another location.

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

* Finds all .txt files inside the given path.

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

* Owner u executes permission, without changing other permissions.

1. **echo $PATH**

* Display the systems PATH variable.

**Part B**

**Identify True or False:**

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

* **True**

1. **mv** is used to move files and directories.

* **True**

1. **cd** is used to copy files and directories.

* **False**
* ***Correct* -cd used to change directory.**

1. **pwd** stands for "print working directory" and displays the current directory.

* True

1. **grep** is used to search for patterns in files.

* True

1. **chmod 755 file.txt** gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

* True

1. **mkdir -p directory1/directory2** creates nested directories, creating directory2 inside directory1if directory1 does not exist.

* True

1. **rm -rf file.txt** deletes a file forcefully without confirmation.

* True

**Identify the Incorrect Commands:**

1. **chmodx** is used to change file permissions.

* Incorrect command
* *Correct command*: correct command **chmod.**

1. **cpy** is used to copy files and directories.

* Incorrect command
* *Correct command*: the correct command is **cp.**

1. **mkfile** is used to create a new file.

* Incorrect command
* *Correct command****:* touch** filename.

1. **catx** is used to concatenate files.

* **Incorrect command.**
* *Correct command:* is **cat**

1. **rn** is used to rename files.

* **Incorrect command**
* *Correct command:* is mv

**Part C**

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

**Commands:** 1. vi hello.sh

2. Insert text into the vi Editor press Esc + I

3. save and exit: Esc + :wq

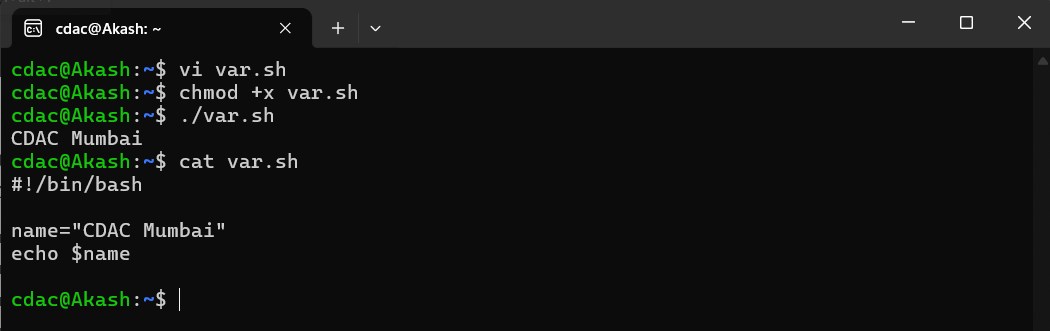
4.chmod +x hello.sh

5. ./hello.sh



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

* vi var.sh
* chmod +x var.sh
* ./var.sh
* #!/bin/bash
* name="CDAC Mumbai"
* echo $name



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

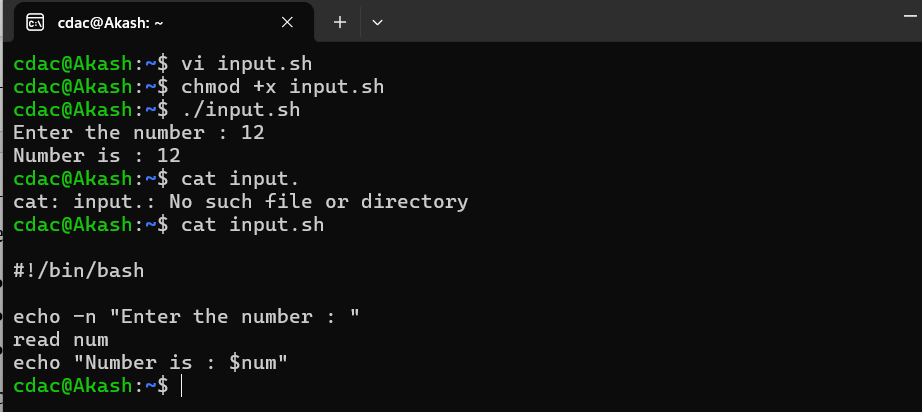
* vi input.sh
* chmod +x input.sh
* ./input.sh

#!/bin/bash

echo -n "Enter the number : "

read num

echo "Number is : $num"



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

* vi add.sh
* chmod +x add.sh
* ./add.sh

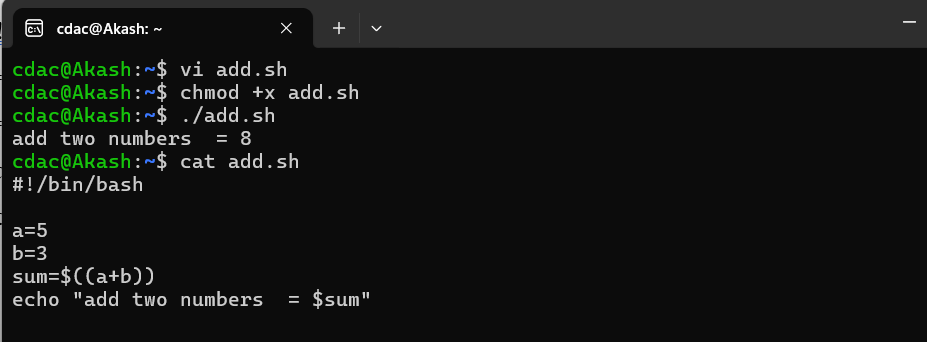
#!/bin/bash

a=5

b=3

sum=$((a+b))

echo "add two numbers = $sum"



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

* vi even.sh
* chmod +x even.sh
* ./even.sh

#!/bin/bash

echo -n "Enter number: "

read num

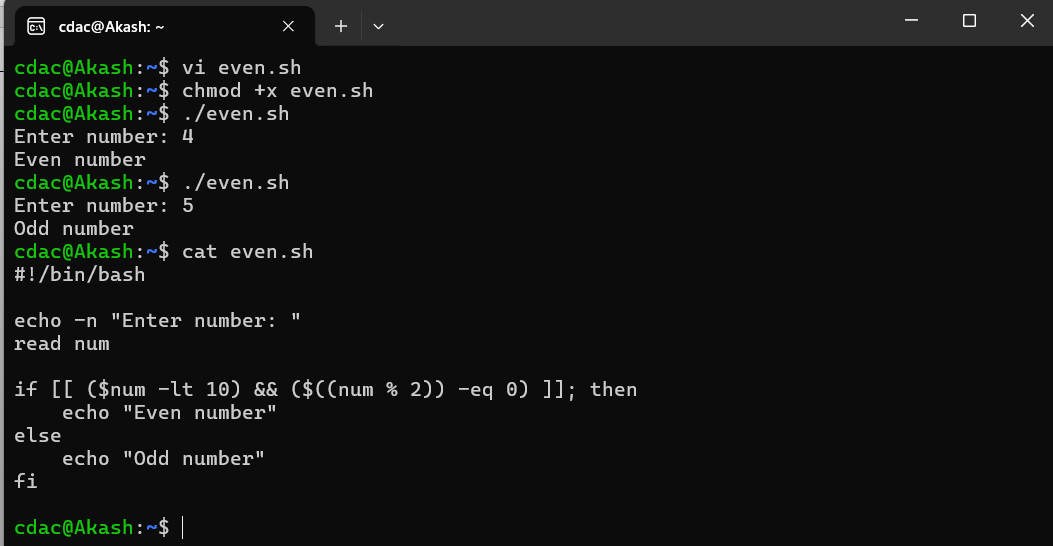
if [[ ($num -lt 10) && ($((num % 2)) -eq 0) ]]; then

echo "Even number"

else

echo "Odd number"

fi



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

* vi loop.sh
* chmod +x loop.sh
* ./loop.sh

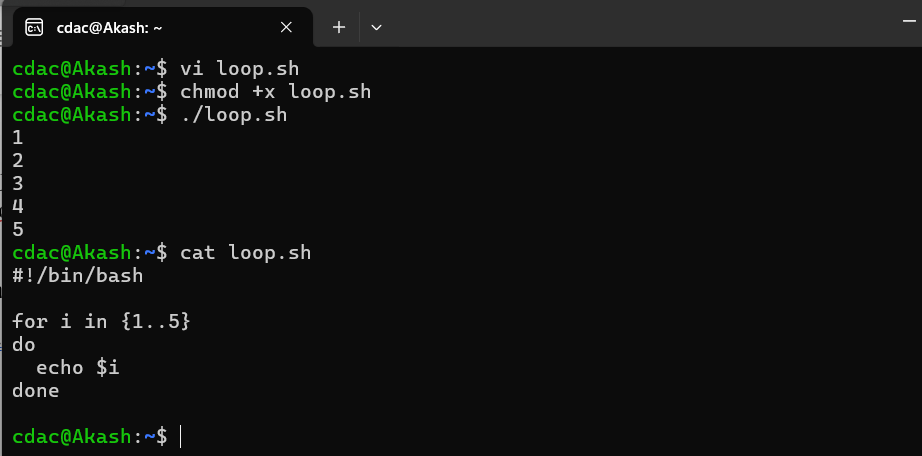
#!/bin/bash

for i in {1..5}

do

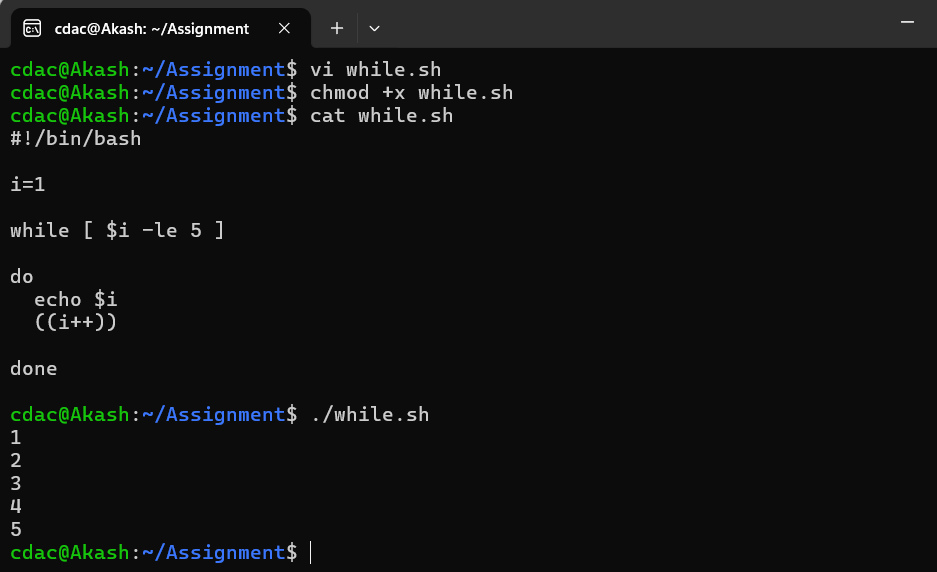
echo $i

done



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

* vi while.sh
* chmod +x while.sh
* cat while.sh
* ./while.sh



**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".

* vi check.sh
* cat check.sh
* chmod +x check.sh
* ./check.sh

#!/bin/bash

if [ -f file.txt ]

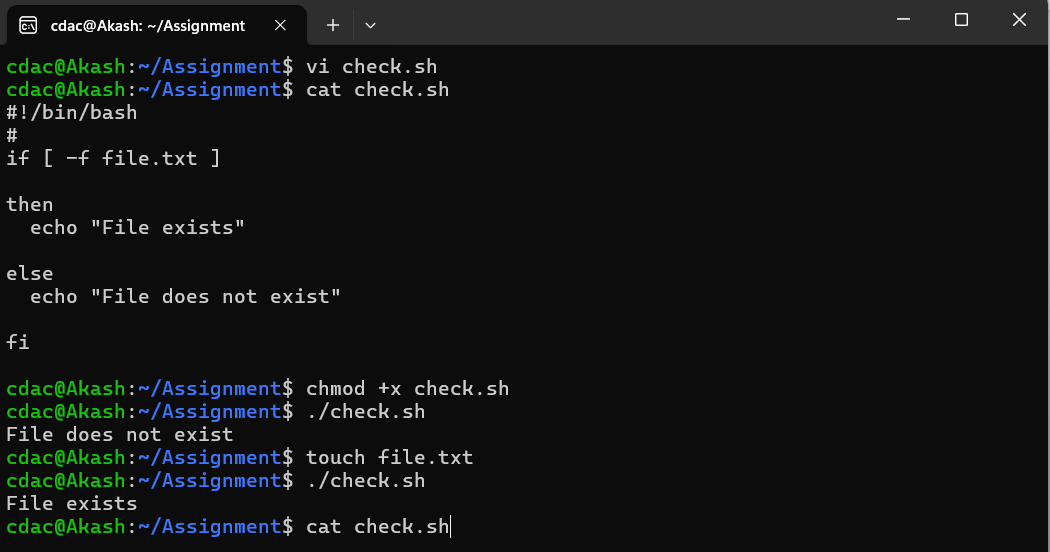
then

echo "File exists"

else

echo "File does not exist"

fi



**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.

#!/bin/bash

echo -n "Enter a number: "

read num

if [ $num -gt 10 ]

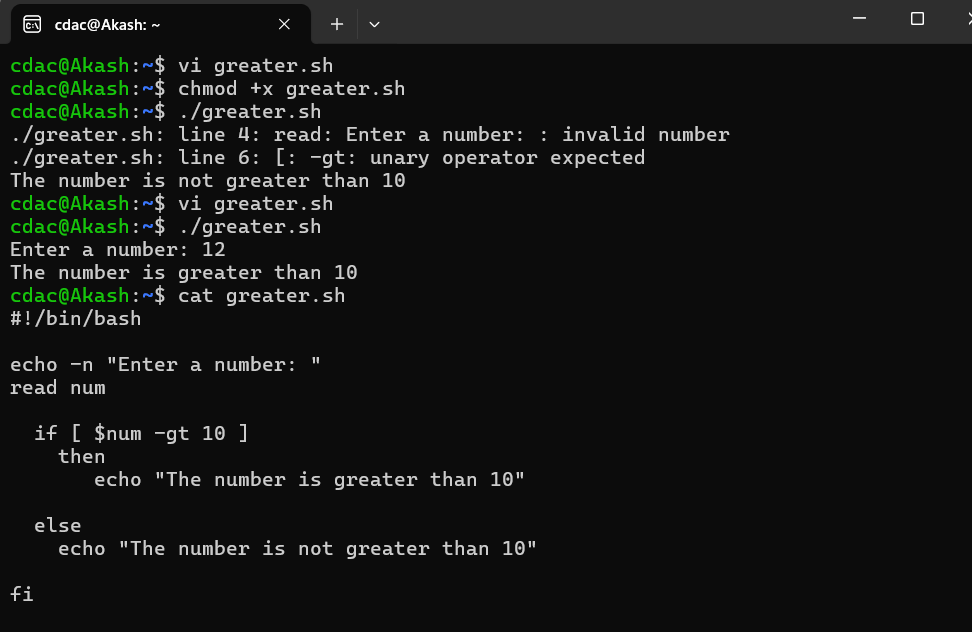
then

echo "The number is greater than 10"

else

echo "The number is not greater than 10"

fi



**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.

#!/bin/bash

for i in {1..5}

do

for j in {1..5}

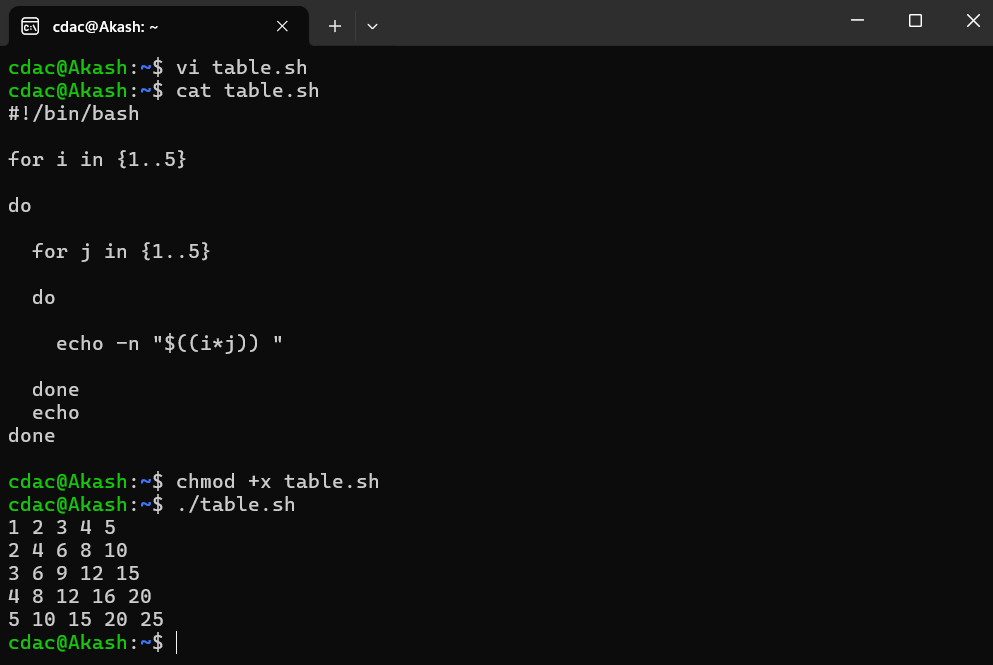
do

echo -n "$((i\*j)) "

done

echo

done



**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.

#!/bin/bash

while true

do

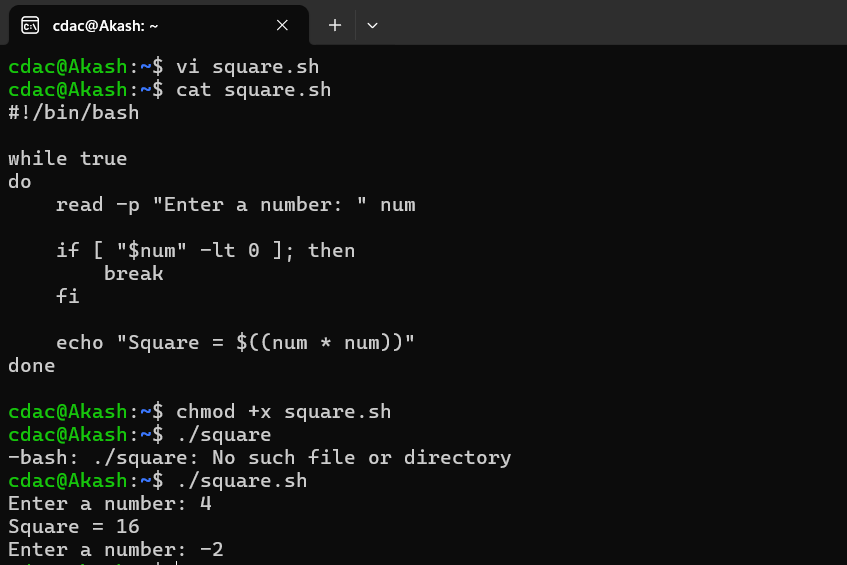
read -p "Enter a number: " num

if [ "$num" -lt 0 ]; then

break

fi

echo "Square = $((num \* num))"

done

**Part E**

