

Name: Shriniwas Vasant Pawar

Center: Kharghar

Email: pawar.shriniwas26@gmail.com

CDAC MUMBAI

Concepts of Operating System

Assignment 2

Part A

What will the following commands do?

1. echo "Hello, World!"

Ans: This command prints “Hello World!” On console.

2. name="Productive"

Ans: This command create a variable “name” with value as “Productive”.

3. touch file.txt

Ans: This command creates a new file with name “file.txt”.

4. ls -a

Ans: This command lists all files and directories including hidden files and directories also.

5. rm file.txt

Ans: This command is used to remove a file with “file.txt” file name.

6. cp file1.txt file2.txt

Ans: This command copies all the contents of “file1.txt” and creates a new file with name “file2.txt” and paste into it.

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

Ans: This command moves “file.txt” file to “/path/to/directory” directory position.

8. chmod 755 script.sh

Ans: This command set permission of “script.sh” file to read,write and execute permission to owner of that file and set read and write permission to group and other users.

9. grep "pattern" file.txt

Ans: This command find a line which contains “pattern” string in “file.txt” and print that line on console.

10. kill PID

Ans: This command kills the process with the mentioned process id or send TERM signal to specified process.

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

Ans: This command firstly create “mydir” directory and then create a file named “file.txt” then adds content “Hello, World!” to it. and after that display “Hello, World!” to the console which is the content of “file.txt” file.

12. ls -l | grep ".txt"

Ans: This command lists out all files with detailed information and end with “.txt” in current directory.

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

Ans: This command displays sorted unique words content of “file1.txt” and “file2.txt” on the console.

14. ls -l | grep "^d"

Ans: This command lists out all directories in the current directory with detailed information.

15. grep -r "pattern" /path/to/directory/

Ans: This command is used to search for string “pattern” recursively within a given directory and its subdirectories.

16. cat file1.txt file2.txt | sort | uniq -d

Ans: This command combines content of “file1.txt” and “file2.txt” then sorts it and displays only duplicated lines in the both files.

17. chmod 644 file.txt

Ans: This command set permission of “file.txt” file as read, write for owner and read permissions to group and other users.

18. cp -r source_directory destination_directory

Ans: This command copies all files, directories, subdirectories and its files from **source_directory** to **destination_directory**.

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

Ans: This command list outs all files ends with “.txt” in its directories and subdirectories.

20. chmod u+x file.txt

Ans: This command sets execute permission to owner without affecting other permissions.

21. echo \$PATH

Ans: This command is used to display the current value of the PATH environment variable.

Part B

Identify True or False:

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

Identify the Incorrect Commands:

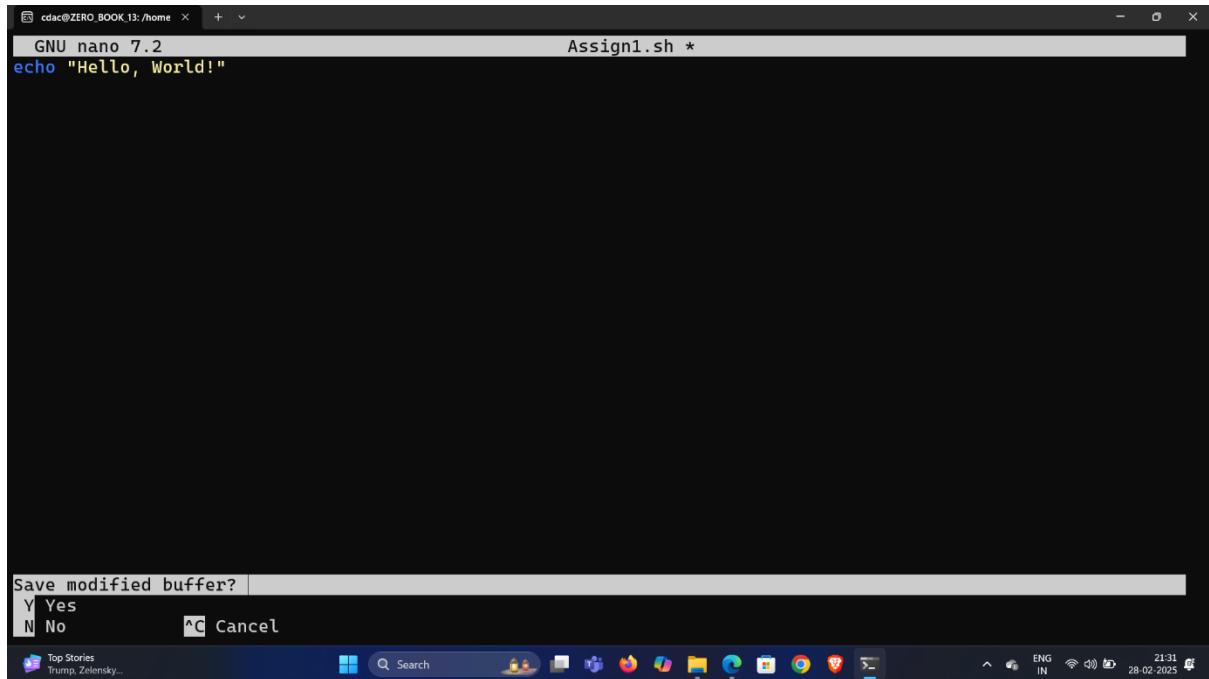
1. **chmodx** is used to change file permissions.
Ans: Incorrect: chmodx is not a valid command. The correct command to change file permissions is chmod.
2. **cpy** is used to copy files and directories.
Ans: Incorrect: cpy is not a valid command. The correct command to copy files and directories is cp.
3. **mkfile** is used to create a new file.
Ans: Incorrect: mkfile is not a valid Linux command to create a new file. You can use touch to create an empty file.
4. **catx** is used to concatenate files.
Ans: Incorrect: catx is not a valid command. The correct command to concatenate files is cat.

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

Ans: Incorrect: **rn** is not a valid command. The correct command to rename files is **mv**.

Part C

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



cdac@ZERO_BOOK_13: /home

GNU nano 7.2 Assign1.sh *

```
echo "Hello, World!"
```

Save modified buffer? |

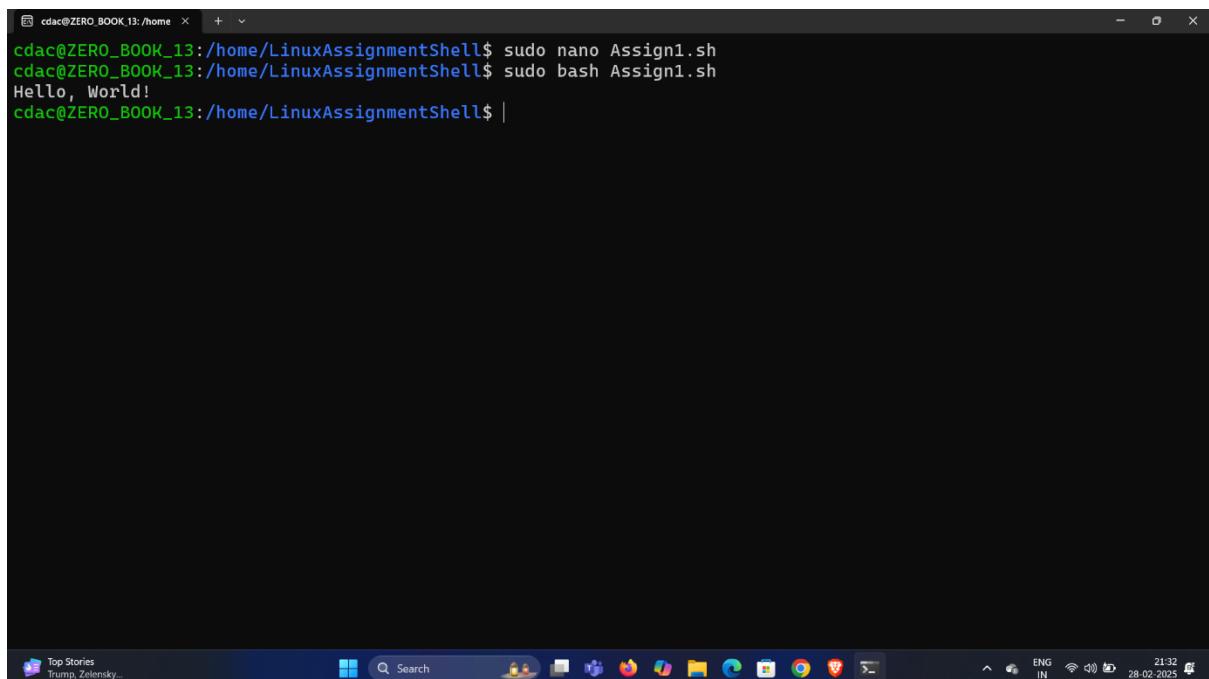
Y Yes

N No

^C Cancel

Top Stories Trump, Zelensky... Search

21:31 28-02-2025

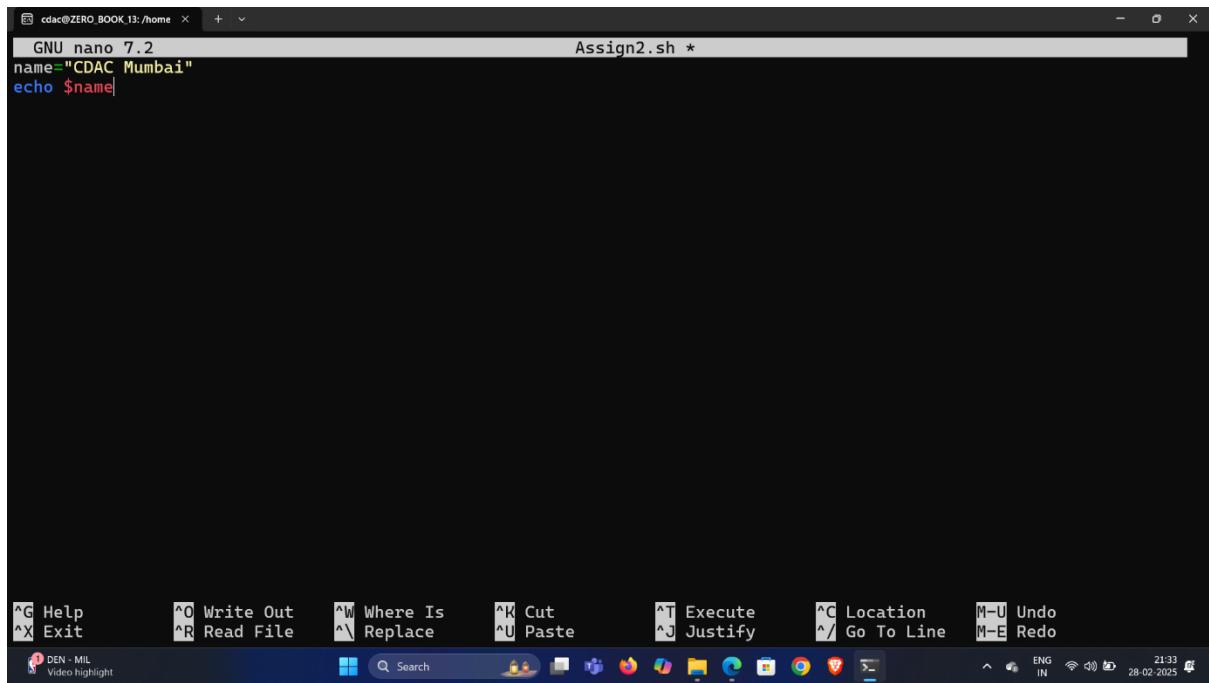


```
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo nano Assign1.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo bash Assign1.sh
Hello, World!
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ |
```

Top Stories Trump, Zelensky... Search

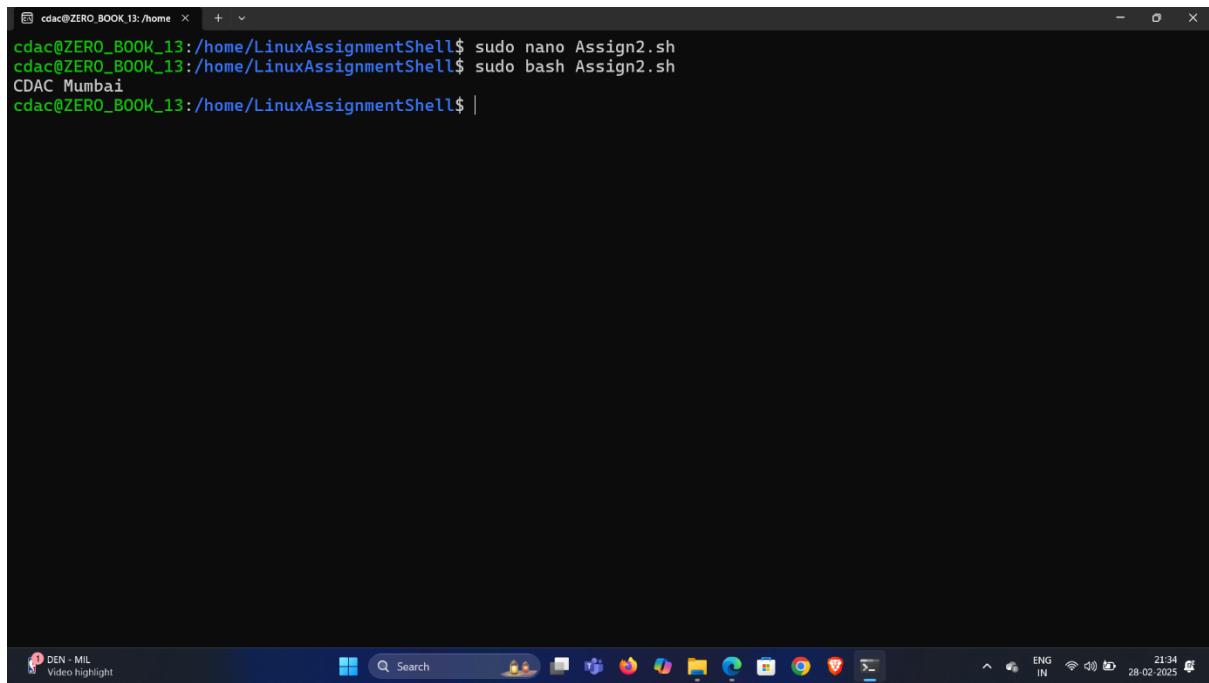
21:32 28-02-2025

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



```
cdac@ZERO_BOOK_13:/home          + - x
GNU nano 7.2                         Assign2.sh *
name="CDAC Mumbai"
echo $name

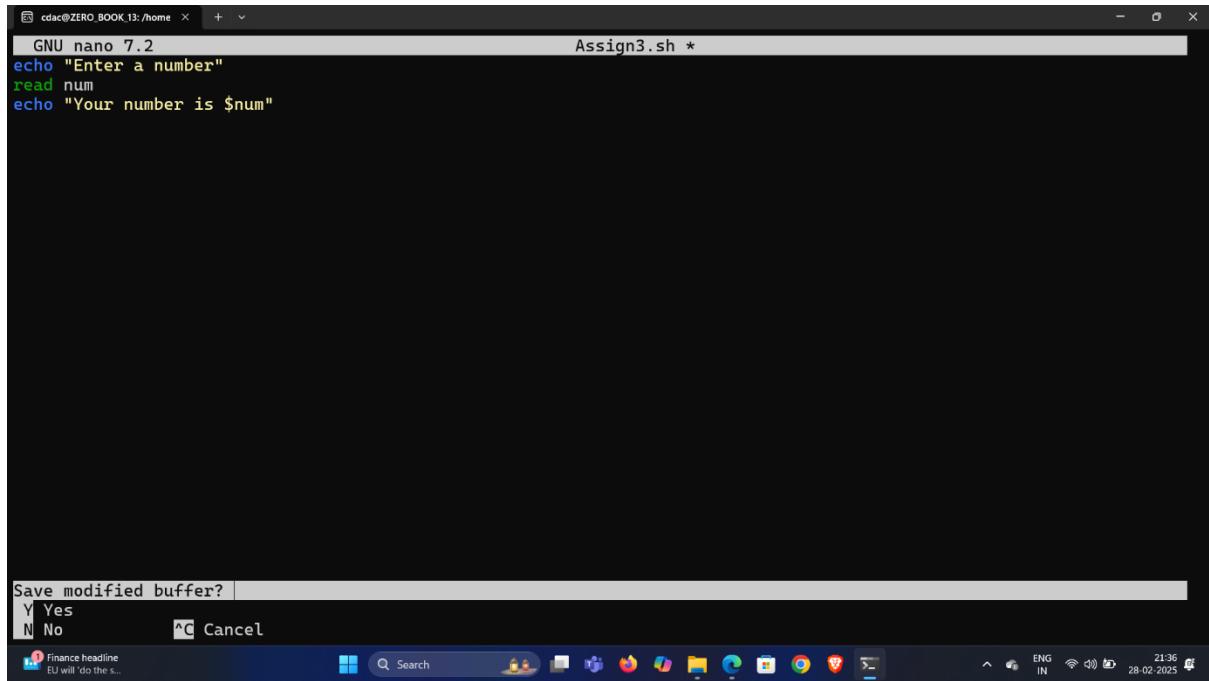
^G Help      ^O Write Out      ^W Where Is      ^K Cut      ^T Execute      ^C Location      M-U Undo
^X Exit      ^R Read File      ^\ Replace      ^U Paste      ^J Justify      ^/ Go To Line      M-E Redo
DEN - MIL  Video highlight      Search      Home      File      Edit      View      Insert      Tools      Help      21:33
28-02-2025
```



```
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo nano Assign2.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo bash Assign2.sh
CDAC Mumbai
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ |
```

```
DEN - MIL  Video highlight      Search      Home      File      Edit      View      Insert      Tools      Help      21:34
28-02-2025
```

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



cdac@ZERO_BOOK_13:/home

GNU nano 7.2 Assign3.sh *

```
echo "Enter a number"
read num
echo "Your number is $num"
```

Save modified buffer?

Yes

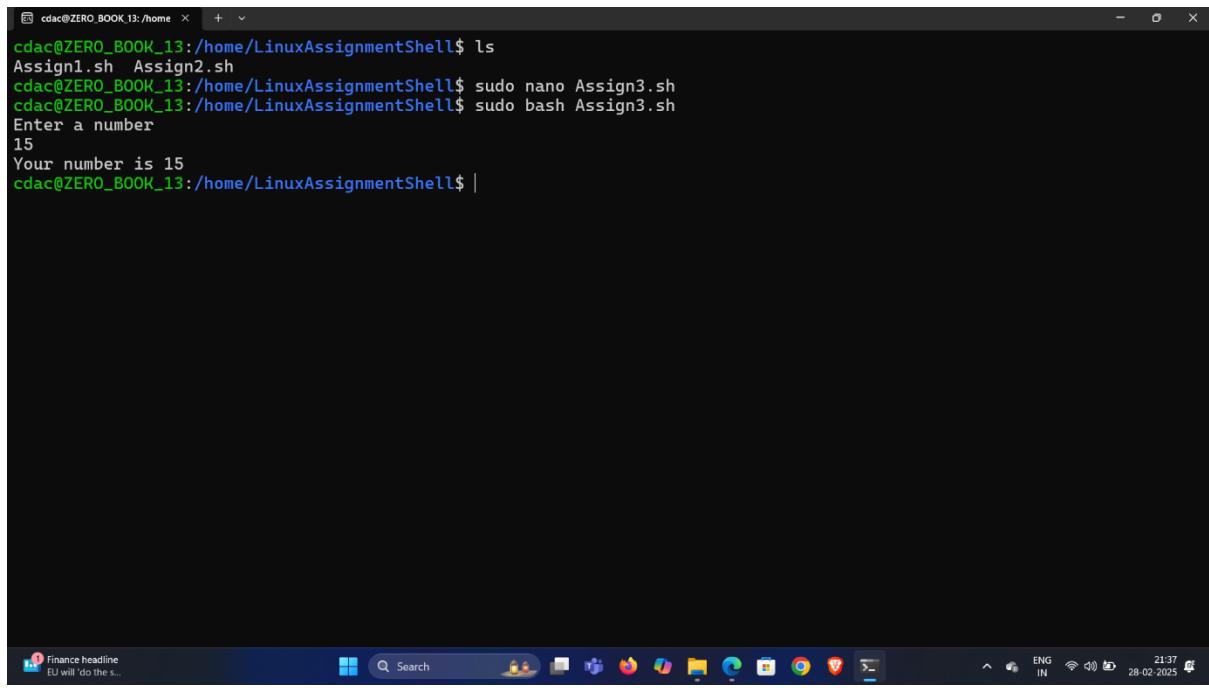
No

Cancel

Finance headline

EU will 'do the s...

21:36 28.02.2025



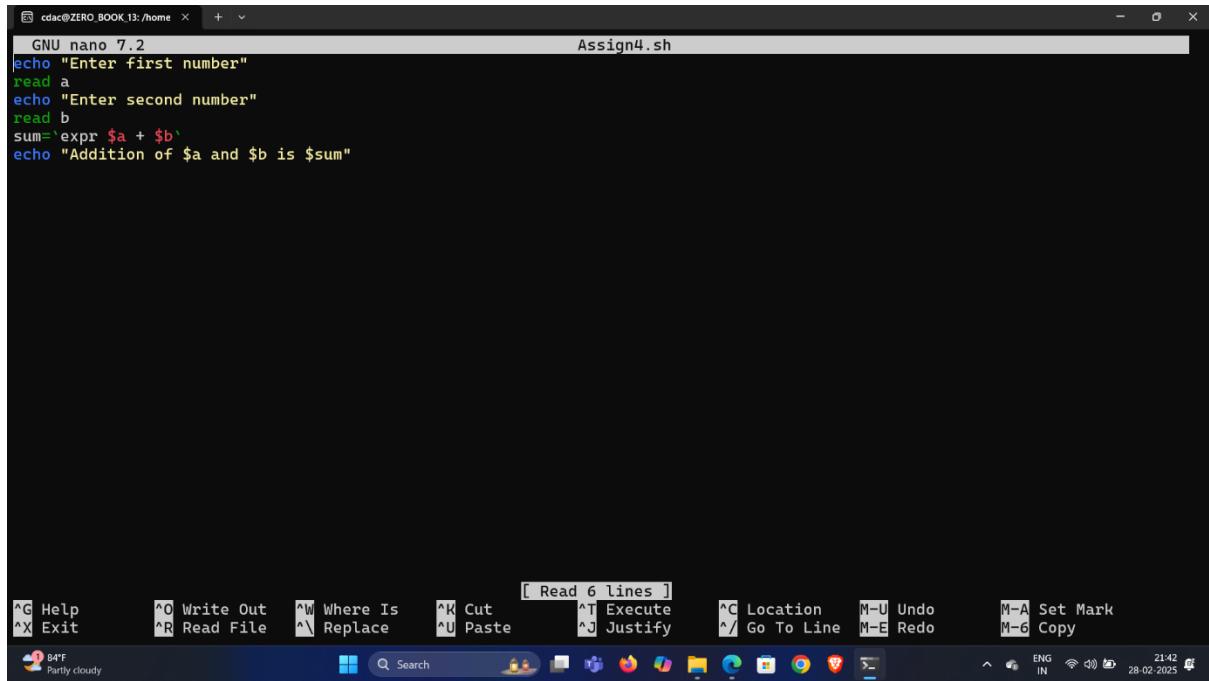
```
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ ls
Assign1.sh Assign2.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo nano Assign3.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo bash Assign3.sh
Enter a number
15
Your number is 15
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ |
```

Finance headline

EU will 'do the s...

21:37 28.02.2025

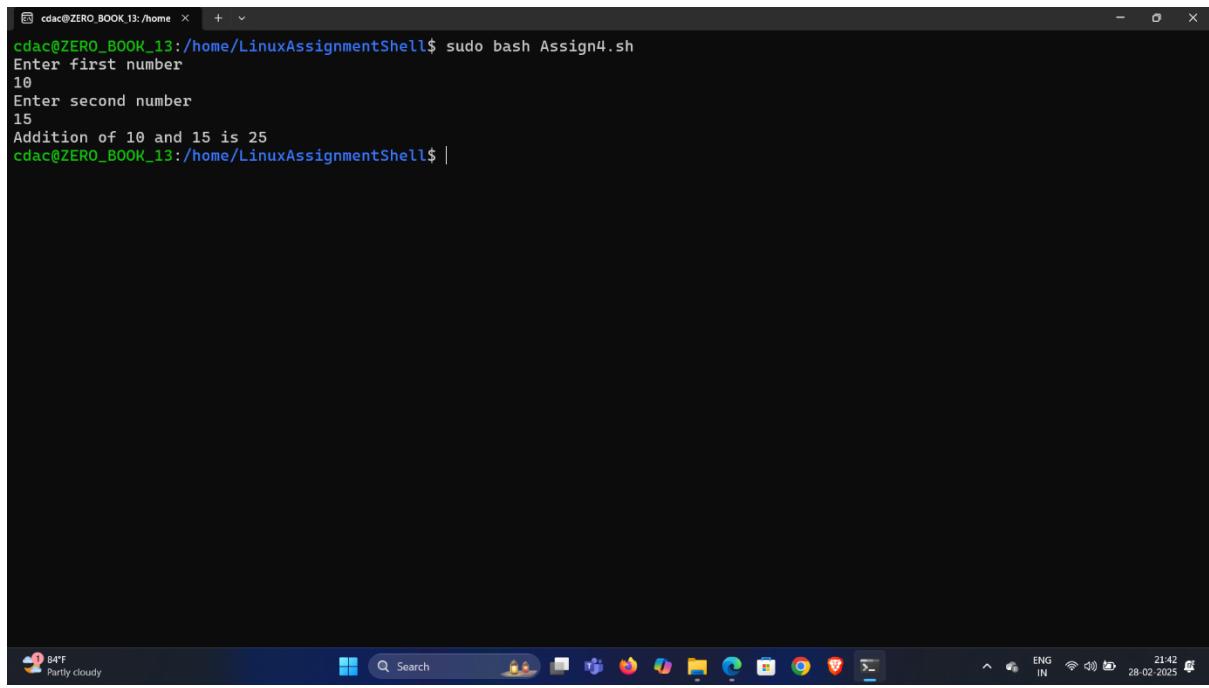
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.



```
GNU nano 7.2                               Assign4.sh
echo "Enter first number"
read a
echo "Enter second number"
read b
sum=`expr $a + $b`
echo "Addition of $a and $b is $sum"
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-G Copy
84°F Partly cloudy

21:42 28.02.2025

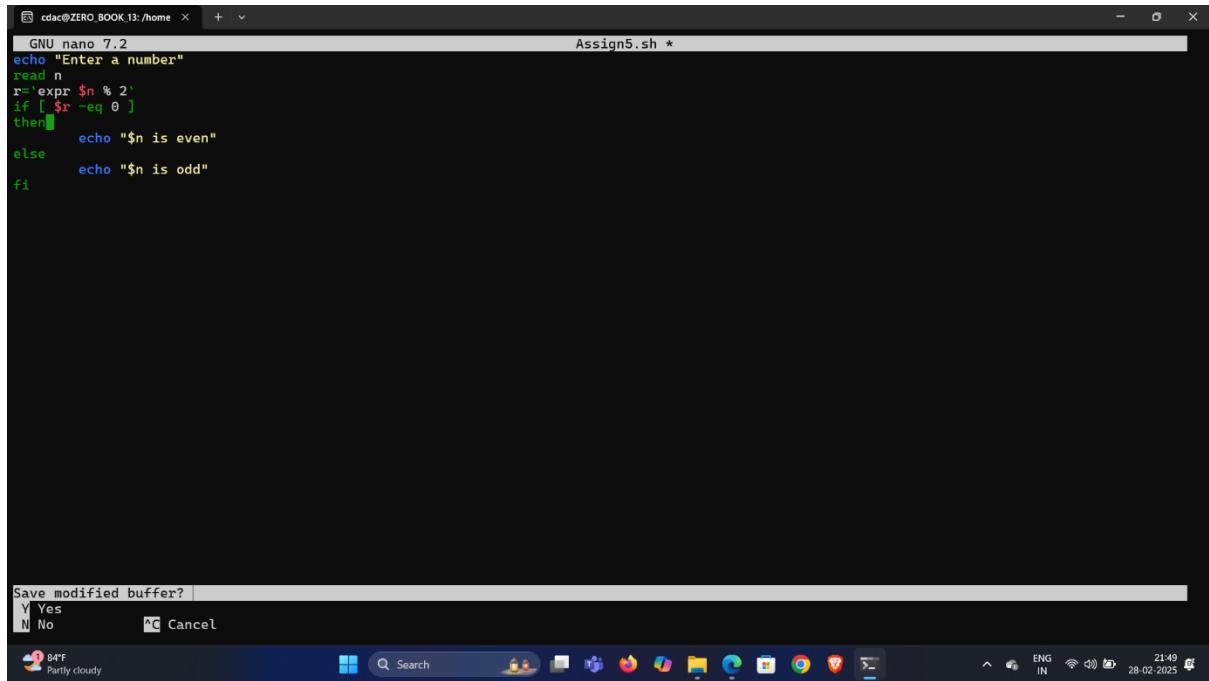


```
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo bash Assign4.sh
Enter first number
10
Enter second number
15
Addition of 10 and 15 is 25
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ |
```

84°F Partly cloudy

21:42 28.02.2025

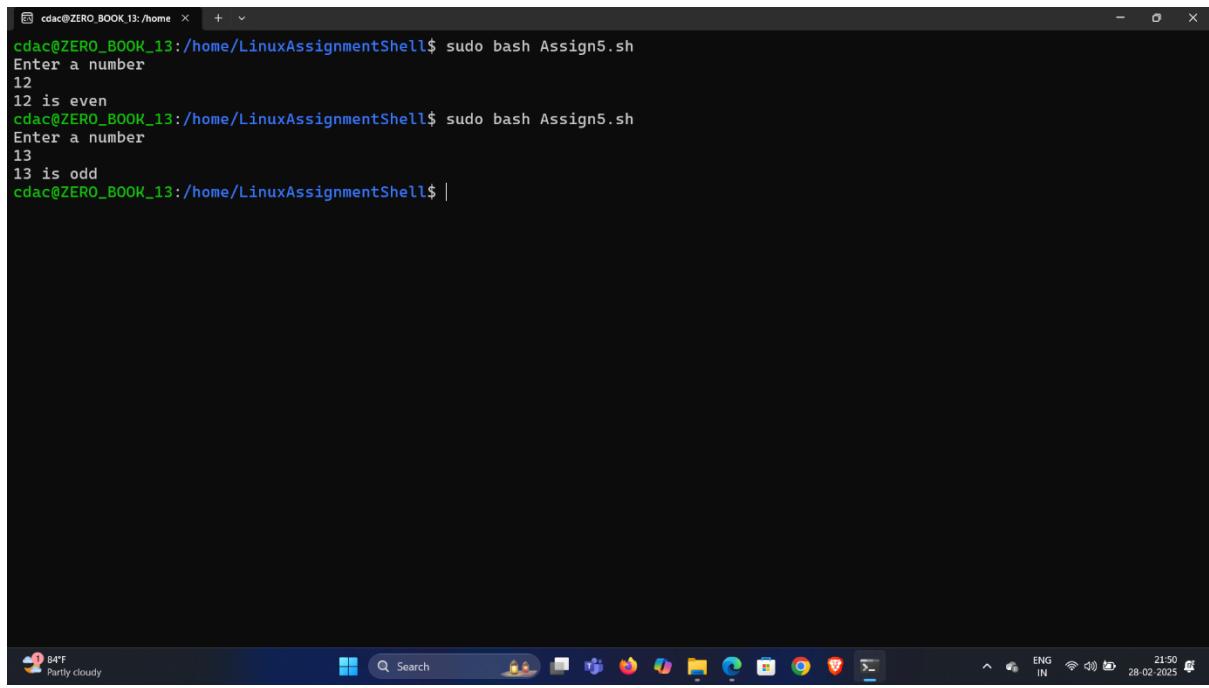
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".



```
GNU nano 7.2                                         Assign5.sh *
echo "Enter a number"
read n
r=$(expr $n % 2)
if [ $r -eq 0 ]
then
    echo "$n is even"
else
    echo "$n is odd"
fi

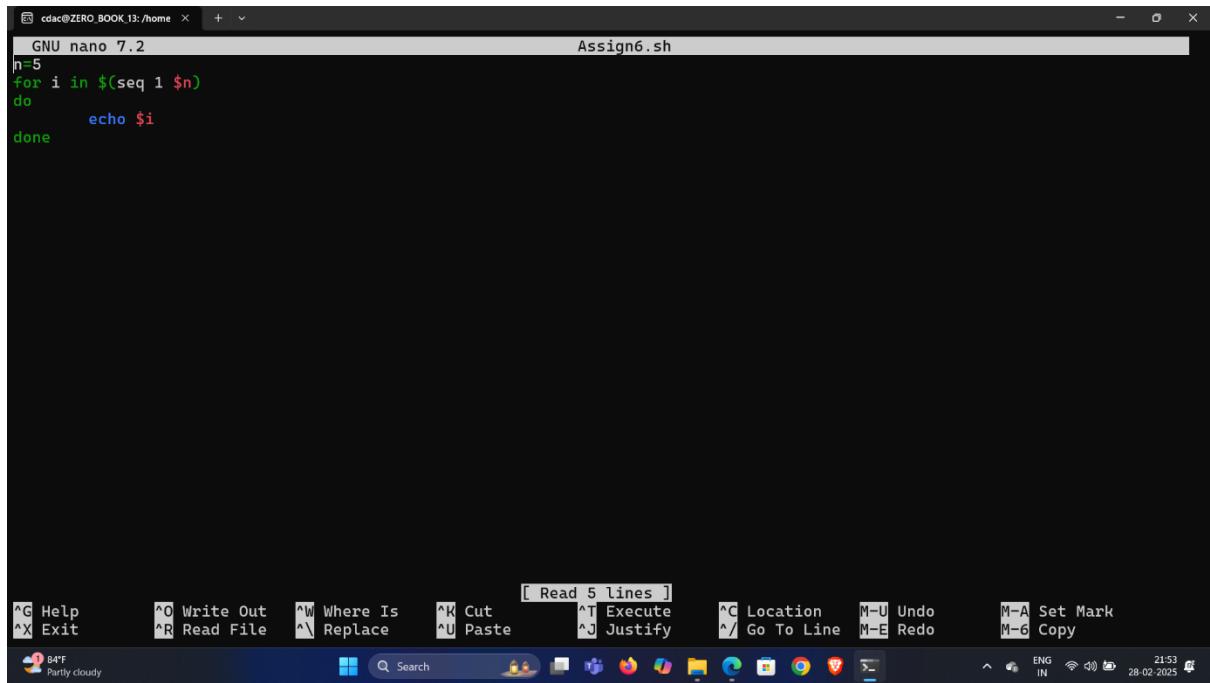
Save modified buffer? |
```

Y Yes
N No **Cancel**



```
cdac@ZERO_BOOK_13:/home                                         Assign5.sh
Enter a number
12
12 is even
cdac@ZERO_BOOK_13:/home                                         Assign5.sh
Enter a number
13
13 is odd
cdac@ZERO_BOOK_13:/home
```

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



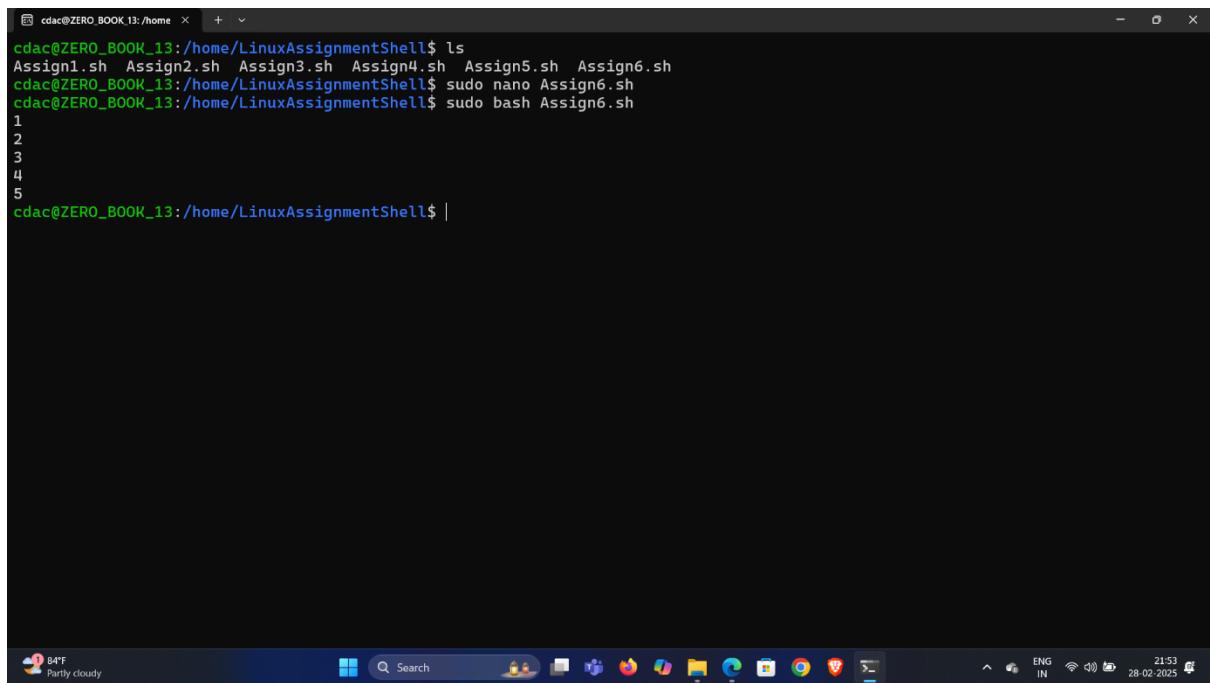
cdac@ZERO_BOOK_13: /home

```
GNU nano 7.2                                     Assign6.sh
n=5
for i in $(seq 1 $n)
do
    echo $i
done
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy
[Read 5 lines]

84°F Partly cloudy

21:53 28-02-2025



```
cdac@ZERO_BOOK_13: /home/LinuxAssignmentShell$ ls
Assign1.sh Assign2.sh Assign3.sh Assign4.sh Assign5.sh Assign6.sh
cdac@ZERO_BOOK_13: /home/LinuxAssignmentShell$ sudo nano Assign6.sh
cdac@ZERO_BOOK_13: /home/LinuxAssignmentShell$ sudo bash Assign6.sh
```

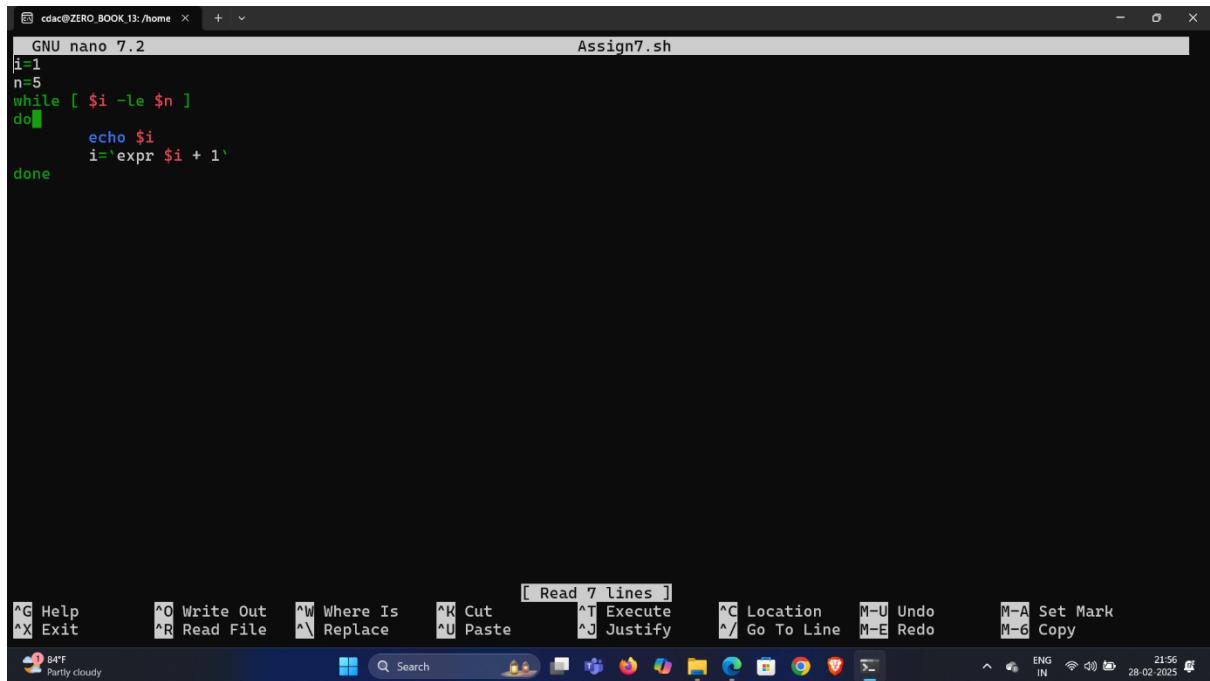
1
2
3
4
5

```
cdac@ZERO_BOOK_13: /home/LinuxAssignmentShell$ |
```

84°F Partly cloudy

21:53 28-02-2025

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

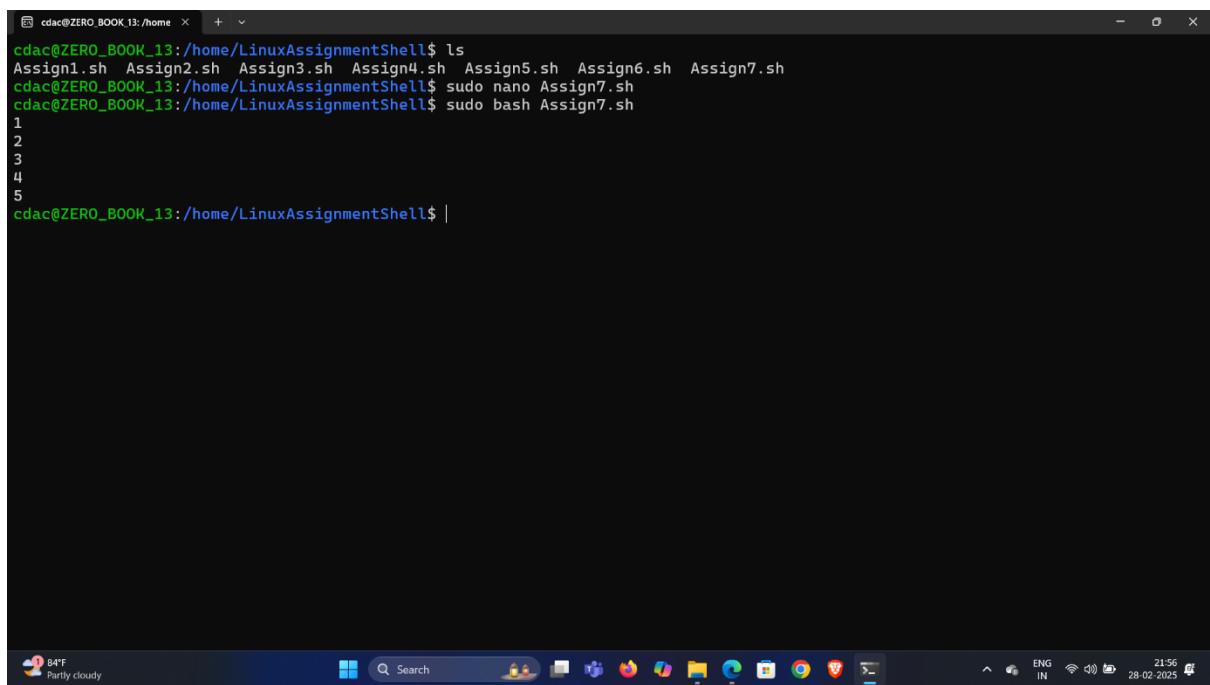


```
GNU nano 7.2                               Assign7.sh
i=1
n=5
while [ $i -le $n ]
do
    echo $i
    i=`expr $i + 1`
done
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy
[Read 7 lines]

84°F Partly cloudy

21:56 28-02-2025

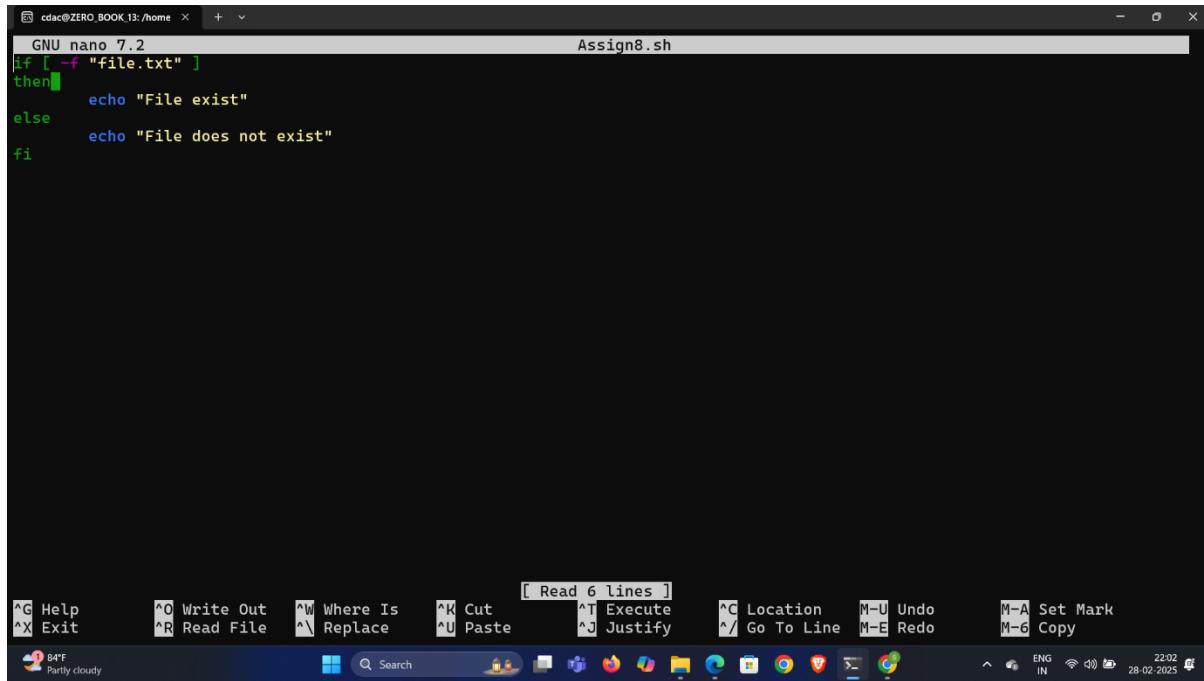


```
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ ls
Assign1.sh Assign2.sh Assign3.sh Assign4.sh Assign5.sh Assign6.sh Assign7.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo nano Assign7.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo bash Assign7.sh
1
2
3
4
5
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ |
```

84°F Partly cloudy

21:56 28-02-2025

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".



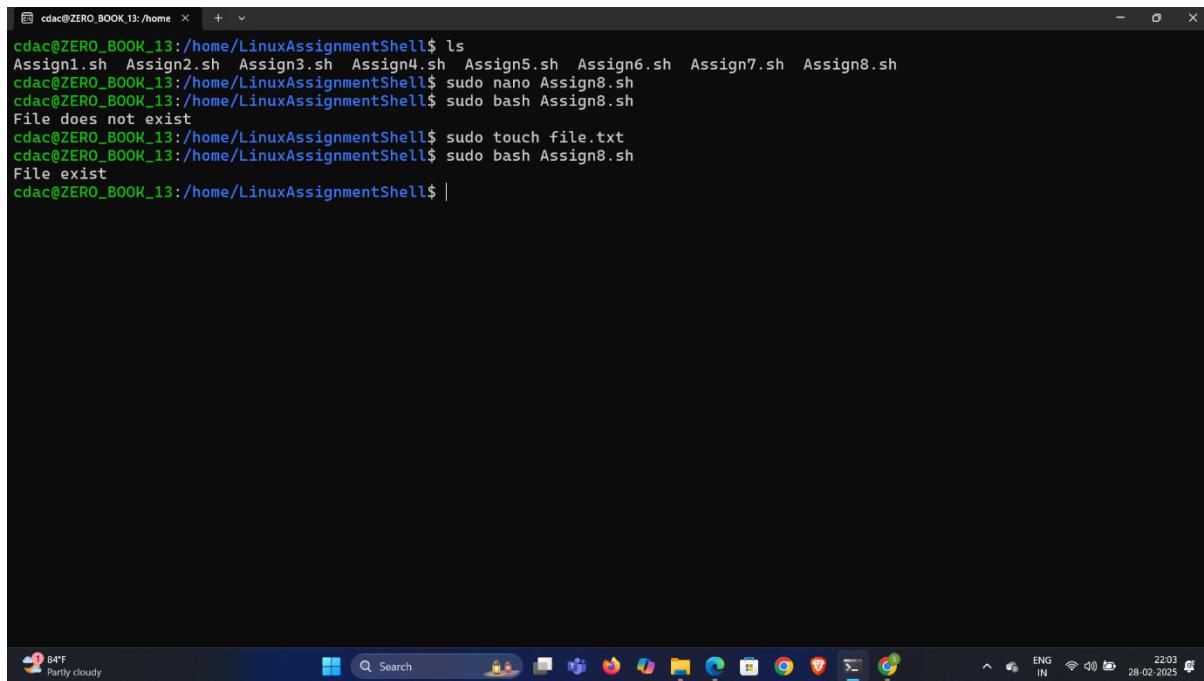
cdac@ZERO_BOOK_13:~/home

```
GNU nano 7.2                                     Assign8.sh
if [ -f "file.txt" ]
then
    echo "File exist"
else
    echo "File does not exist"
fi
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-G Copy
84°F
Party cloudy

22:02 28.02.2025

This screenshot shows a terminal window titled "Assign8.sh" containing a shell script. The script uses an if-then-else construct to check if a file named "file.txt" exists in the current directory. If it does, it prints "File exist"; otherwise, it prints "File does not exist". The terminal is running on a Linux system, as indicated by the nano editor and the command-line interface.



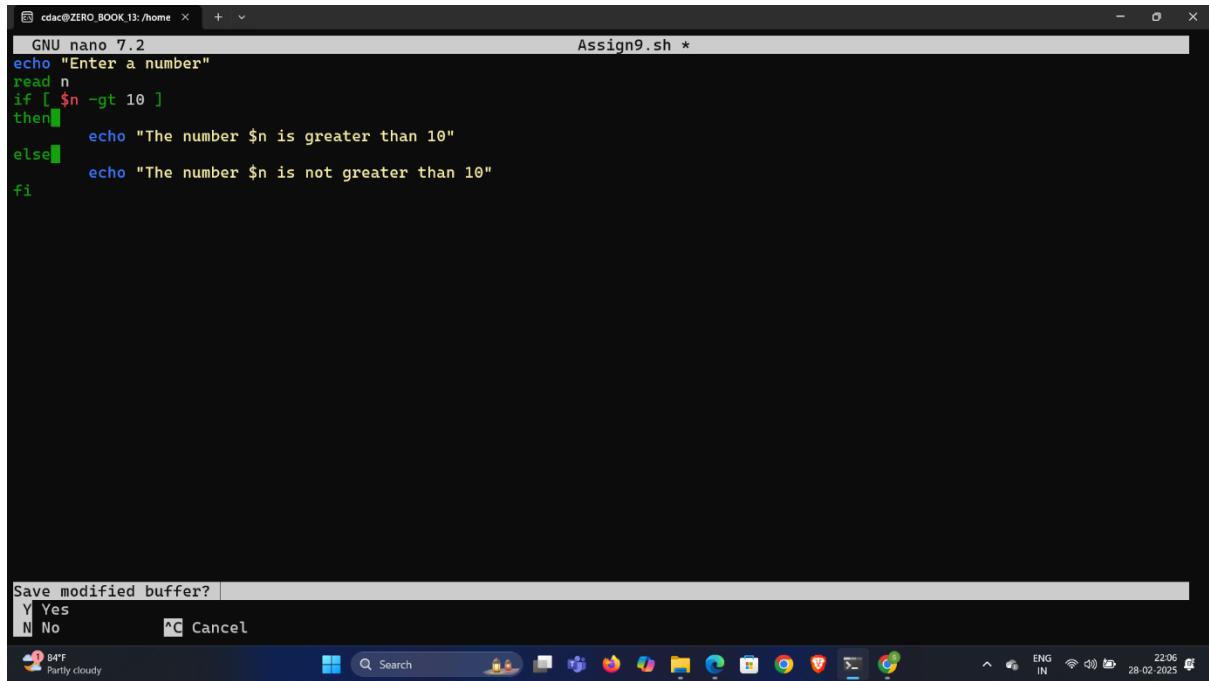
```
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ ls
Assign1.sh Assign2.sh Assign3.sh Assign4.sh Assign5.sh Assign6.sh Assign7.sh Assign8.sh
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ sudo nano Assign8.sh
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ sudo bash Assign8.sh
File does not exist
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ sudo touch file.txt
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ sudo bash Assign8.sh
File exist
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ |
```

84°F
Party cloudy

22:03 28.02.2025

This screenshot shows the same terminal window after the script has been executed. The user runs "ls" to list files, then "sudo nano Assign8.sh" to edit the script. When run with "sudo bash Assign8.sh", the script prints "File does not exist" because no "file.txt" file exists. The user then runs "sudo touch file.txt" to create the file, and when the script is run again, it prints "File exist". The terminal window shows the script's content and the user's interactions.

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.



cdac@ZERO_BOOK_13:~/home

```
GNU nano 7.2                                     Assign9.sh *
```

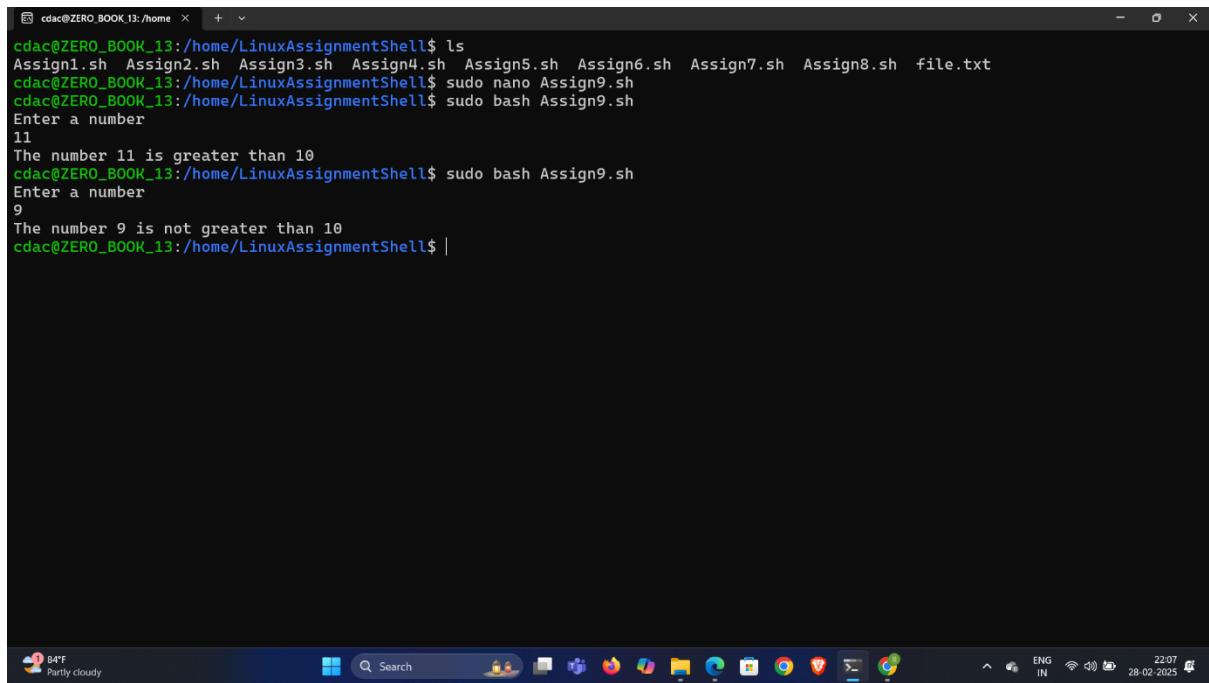
```
echo "Enter a number"
read n
if [ $n -gt 10 ]
then
    echo "The number $n is greater than 10"
else
    echo "The number $n is not greater than 10"
fi
```

Save modified buffer? |
Y Yes
N No Cancel

84°F Partly cloudy

22:06 28.02.2025

This screenshot shows a terminal window titled 'Assign9.sh *'. It contains a shell script with an 'if' statement that checks if a number is greater than 10. The script uses 'read n' to get input and 'echo' to print messages. A save dialog box is open at the bottom, asking 'Save modified buffer?' with options 'Yes', 'No', and 'Cancel'. The terminal is running on a Linux system with a desktop environment, as indicated by the window title and the taskbar at the bottom.



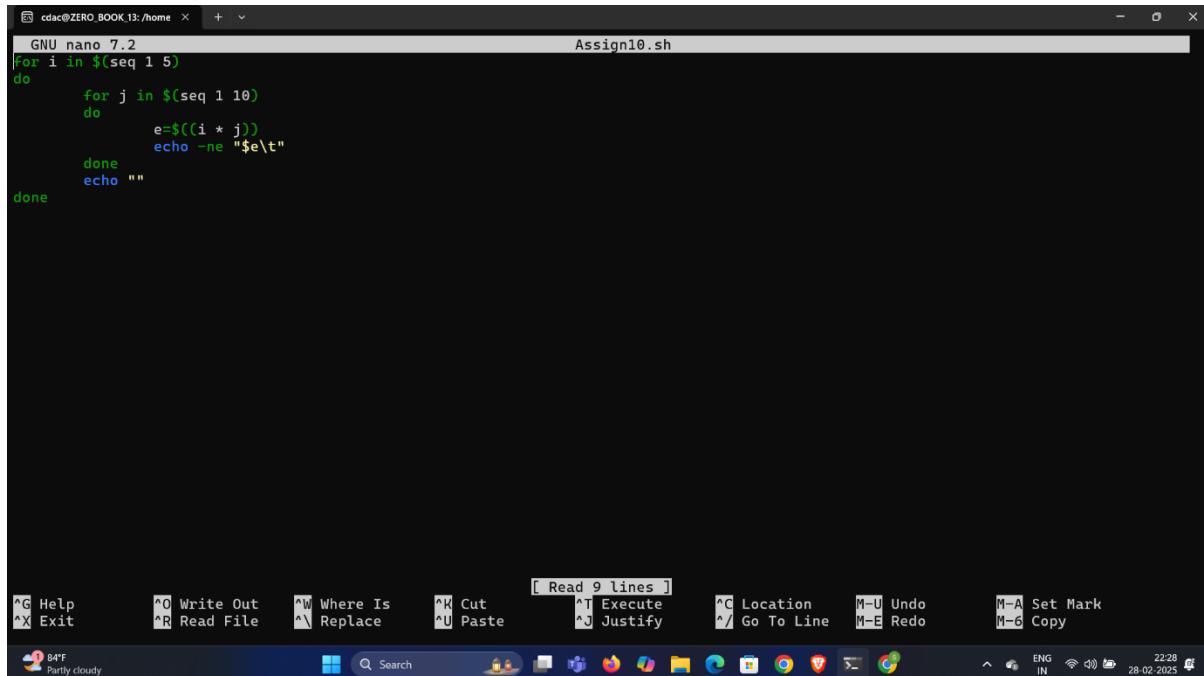
```
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ ls
Assign1.sh Assign2.sh Assign3.sh Assign4.sh Assign5.sh Assign6.sh Assign7.sh Assign8.sh file.txt
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ sudo nano Assign9.sh
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ sudo bash Assign9.sh
Enter a number
11
The number 11 is greater than 10
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ sudo bash Assign9.sh
Enter a number
9
The number 9 is not greater than 10
cdac@ZERO_BOOK_13:~/home/LinuxAssignmentShell$ |
```

84°F Partly cloudy

22:07 28.02.2025

This screenshot shows the terminal window again, this time after running the script. It displays the output of the 'ls' command, the creation of the 'Assign9.sh' file, and its execution. The script prompts for a number and prints whether it is greater than 10. The terminal shows two runs of the script, one for '11' and one for '9'. The desktop environment taskbar is visible at the bottom.

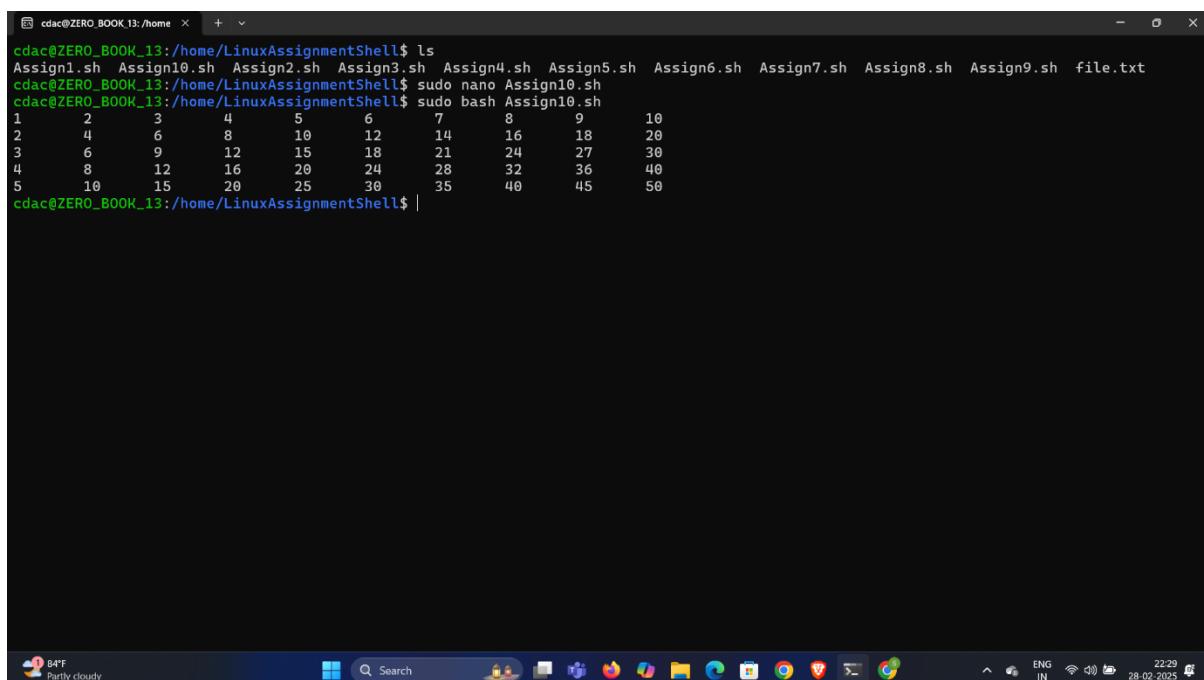
Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.



```
GNU nano 7.2                               Assign10.sh
for i in $(seq 1 5)
do
    for j in $(seq 1 10)
    do
        e=$((i * j))
        echo -ne "$e\t"
    done
    echo ""
done
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo ^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-A Set Mark M-6 Copy

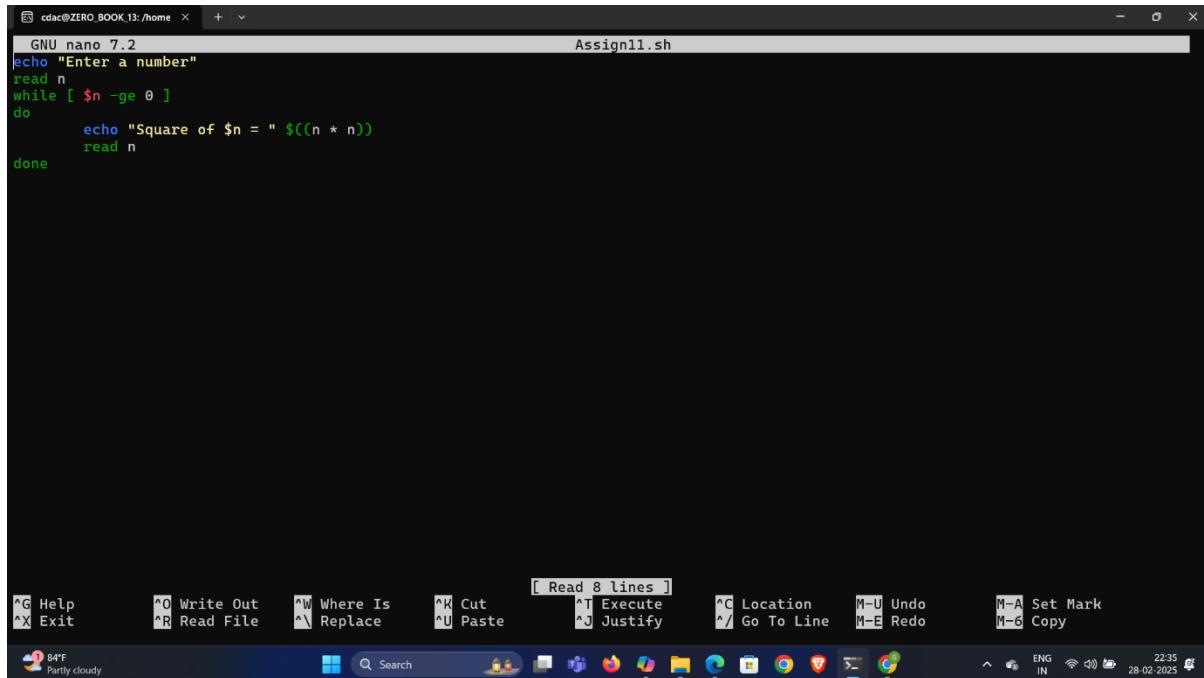
84°F Partly cloudy 28-02-2025 22:28



```
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ ls
Assign1.sh Assign10.sh Assign2.sh Assign3.sh Assign4.sh Assign5.sh Assign6.sh Assign7.sh Assign8.sh Assign9.sh file.txt
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo nano Assign10.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo bash Assign10.sh
1   2   3   4   5   6   7   8   9   10
2   4   6   8   10  12  14  16  18  20
3   6   9   12  15  18  21  24  27  30
4   8   12  16  20  24  28  32  36  40
5   10  15  20  25  30  35  40  45  50
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$
```

84°F Partly cloudy 28-02-2025 22:29

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.



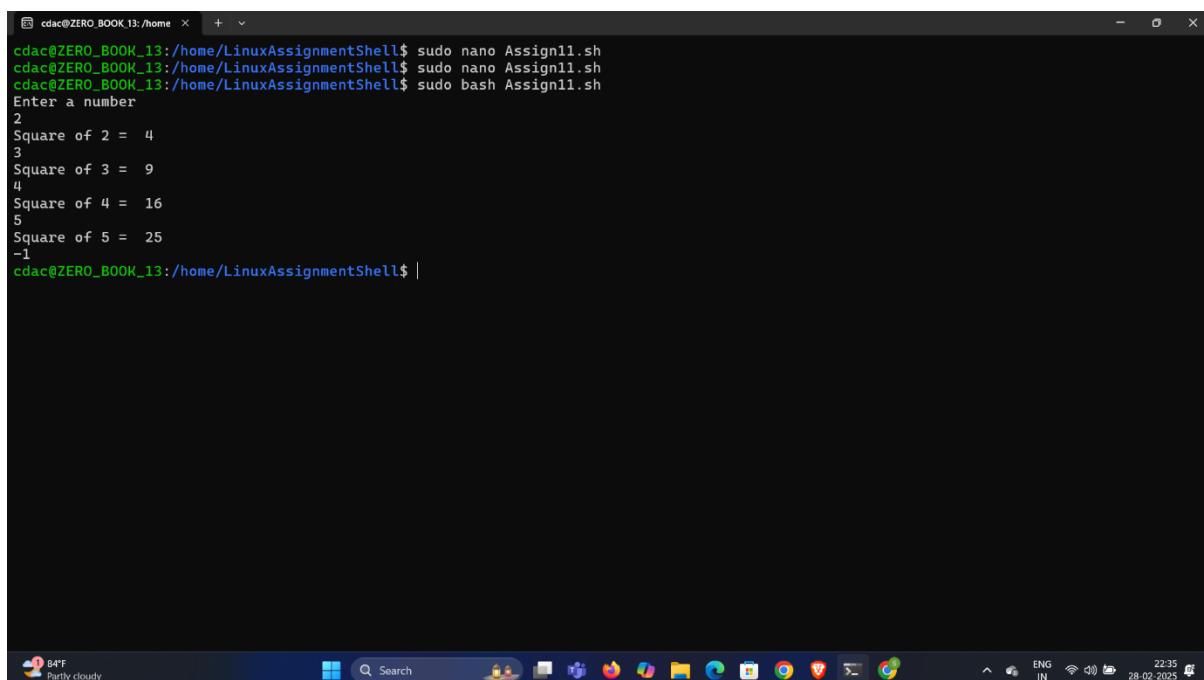
```
GNU nano 7.2                               Assign11.sh
echo "Enter a number"
read n
while [ $n -ge 0 ]
do
    echo "Square of $n = " $((n * n))
    read n
done
```

^G Help ^O Write Out ^W Where Is [Read 8 lines]
^X Exit ^R Read File ^\ Replace ^K Cut ^T Execute ^C Location M-U Undo
^U Paste ^J Justify ^/ Go To Line M-E Redo M-A Set Mark
M-6 Copy

84°F Partly cloudy

Search

28-02-2025 22:35



```
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo nano Assign11.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo nano Assign11.sh
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ sudo bash Assign11.sh
Enter a number
2
Square of 2 = 4
3
Square of 3 = 9
4
Square of 4 = 16
5
Square of 5 = 25
-1
cdac@ZERO_BOOK_13:/home/LinuxAssignmentShell$ |
```

84°F Partly cloudy

Search

28-02-2025 22:35

Assignment - 2



Part E

- 1.) Consider the following processes with arrival times and burst times.

process	Arrival time	Burst time
P ₁	0	5
P ₂	1	3
P ₃	2	6

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

process	Arrival time	Burst time	waiting time
P ₁	0	5	0
P ₂	1	3	3.5
P ₃	2	6	7.8

Grant chart	P ₁	P ₂	P ₃
	0	5	8

$$\therefore \text{Avg waiting time} = \frac{0+5+8}{3}$$

$$= \frac{13}{3}$$

$$\text{Avg waiting time} = 4.33$$

Ques 2) Consider the following processes with arrival time and burst time with

→

Process	Arrival time	Burst time	Waiting time	Turnaround time
P ₁	0	3	0	3
P ₂	1	5	3	8
P ₃	2	1	8	9
P ₄	3	4	13	17

→ calculate average turnaround time using
fairness ratio (0.75) based on SJF scheduling

←

Process	Arrival time	Burst time	Waiting time	Turnaround time
P ₁	0	3	0	3
P ₂	1	5	3	8
P ₃	2	1	8	9
P ₄	3	4	13	17

Grant Chart

P ₁	P ₁	P ₁	P ₁	P ₃	P ₄
0	1	2	3	3	4

avg turn around time = $\frac{0+8+3+4}{4} = \frac{15}{4} = 3.75$

Q3) Consider the following processes with arrival times, burst times and priorities.
(Lower number indicate higher priority)

process	Arrival time	Burst time	Priority	
P ₁	0	6	3	5
P ₂	1	4	1	4
P ₃	2	7	4	7
P ₄	3	2	2	2

Calculate avg waiting time using priority scheduling.

Grant chart

P ₁	P ₂	P ₂	P ₂	P ₂	P ₄	P ₁	P ₃
0	1	2	3	5	7	12	19
5	9	14	21	29	39	57	
15	21	28	35	42	52	67	
35	41	48	55	62	72	87	

Process	Arrival time	Burst time	Priority	Waiting time
P ₁	0	6	3	6
P ₂	1	4	1	0
P ₃	2	7	4	10
P ₄	3	2	2	2

$$\therefore \text{Avg waiting time} = \frac{6+0+10+2}{4} = \frac{18}{4}$$

$$\text{avg waiting time} = 4.5$$

Q4) Consider the following processes with arrival times, burst time and the time quantum for round robin scheduling, is 2 units.

process	Arrival time	burst time
P ₁	0	4
P ₂	1	5
P ₃	2	2
P ₄	3	3

Calculate the average turnaround time using round robin scheduling.

gantt chart:

P ₁	P ₂	P ₃	P ₄	P ₁	P ₂	P ₄	
0	2	4	6	8	10	12	14

~~0 2 4 6 8 10 12 14~~

~~0 2 4 6 8 10 12 14~~

P₁

P₂
1.6

Process	Waiting time	turnaround time
P ₁	6	10
P ₂	9	14
P ₃	2	4
P ₄	7	10

$$\text{avg turnaround time} = \frac{10+14+4+10}{4} = 9$$

$$\text{avg waiting time} = \frac{14+38}{4} = 11$$

$$\text{avg turnaround time} = 9.5$$

avg waiting time = 13.75

when CPU is not idle.

Want chart:

	P ₁	P ₂	P ₃	P ₄	P ₁	P ₂	P ₄
0	2	4	6	8	10	12	13

2 + 6 + 1 = 9.5

waiting time = 3.75

P ₂
14

Process	W.T	FAT	avg waiting time
P ₁	6	10	6.5
P ₂	8	13	8.25
P ₃	2	4	2.5
P ₄	7	10	7.25

$$\therefore \text{Avg FAT} = \frac{10+13+4+10}{4} = \frac{37}{4}$$

$$\text{Avg TAAT} = 9.25$$

Q 5) Consider a program that uses fork() system call to create a child process. Initially, the parent process has a variable x with 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

~~→~~ After

initial value of $x = 5$ in both child and parent process

After the fork command The values of fork are

parent process : $x = 8$

child process : $x = 8$