
PART C

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

Command: --

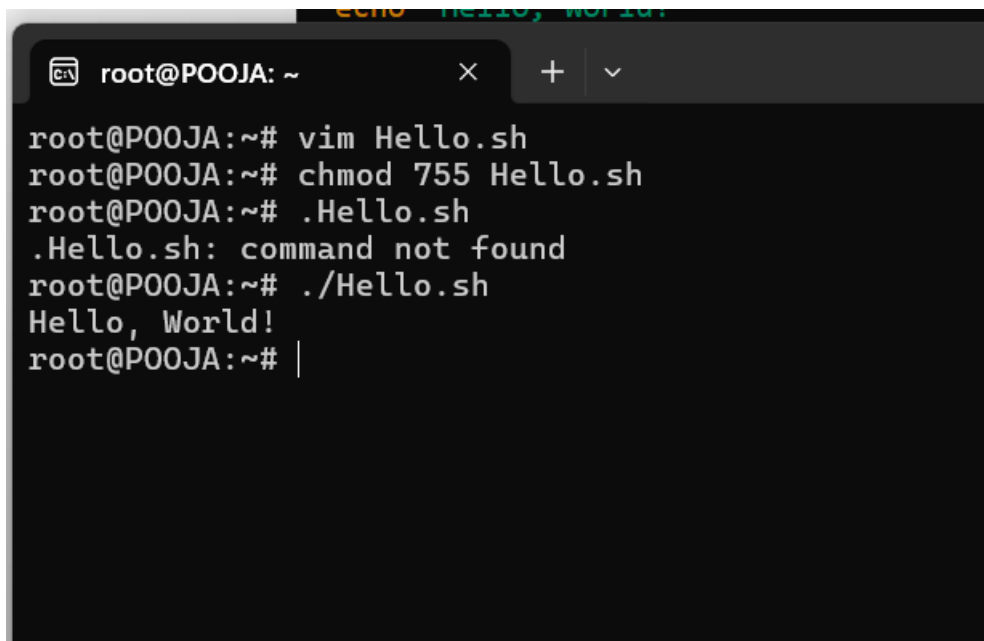
vim hello.sh

I use vim editor to enter data inside the script.

Chmod 775 Hello.sh

I use chmod command to give permission to file access.

./hello.sh this is use for run script.

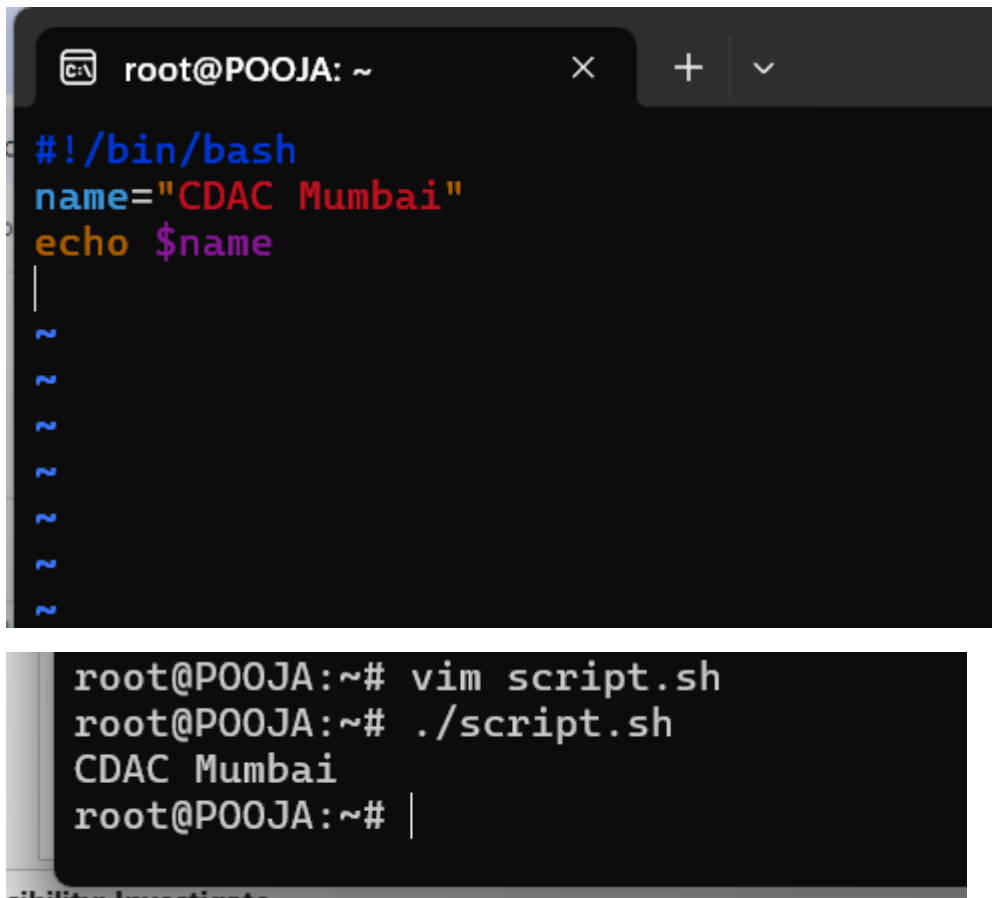
A terminal window titled 'root@POOJA: ~' with standard window controls. The terminal shows the following commands and output:

```
root@POOJA:~# vim Hello.sh
root@POOJA:~# chmod 755 Hello.sh
root@POOJA:~# .Hello.sh
.Hello.sh: command not found
root@POOJA:~# ./Hello.sh
Hello, World!
root@POOJA:~# |
```

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

Command

The variable is assigned with name="CDAC Mumbai" and is printed using echo \$name. The value "CDAC Mumbai" is displayed when the script is run.



The image contains two terminal window screenshots. The top screenshot shows a terminal window titled 'root@POOJA: ~' with a dark background. It displays the following commands and their output: `#!/bin/bash`, `name="CDAC Mumbai"`, `echo $name`, followed by a blank line and then seven lines of the character '2'. The bottom screenshot shows a terminal window with the following commands and output: `root@POOJA:~# vim script.sh`, `root@POOJA:~# ./script.sh`, `CDAC Mumbai`, and `root@POOJA:~#` followed by a cursor.

```
root@POOJA: ~
#!/bin/bash
name="CDAC Mumbai"
echo $name
|
2
2
2
2
2
2
2

root@POOJA:~# vim script.sh
root@POOJA:~# ./script.sh
CDAC Mumbai
root@POOJA:~# |
```

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

Explanation :--

The script uses `read num` to take user input, stores it in the variable `num`, and prints the value with `echo $num`.

```
root@POOJA: ~
#!/bin/bash
echo "Enter the new no."
read num

if ((num%2==0))
then
    echo "Even"
else
    echo "ODD"
fi
~
~
~
```

```
root@POOJA:~# ./new.sh
Enter the new no.
5
ODD
root@POOJA:~# vim new.sh
root@POOJA:~# ./new.sh
Enter the new no.
5
ODD
root@POOJA:~# ./new.sh
Enter the new no.
8
Even
root@POOJA:~# |
```

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

```
root@POOJA: ~
#!/bin/bash

echo "Enter the first number:"
read num1
echo "Enter the second number:"
read num2

sum=$((num1 + num2))

echo "Sum of numbers: $sum"
```

```
root@POOJA:~# vim script1.sh
root@POOJA:~# vim script1.sh
root@POOJA:~# ./script1.sh
Enter the first number:
4
Enter the second number:
5
Sum of numbers: 9
root@POOJA:~# |
```

The script uses `read num` to take user input, stores it in the variable `num`, and prints the value with `echo $num`.

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

The script uses an `if` statement to check if `num % 2 == 0` for evenness, and prints "Even" or "Odd" accordingly.

```
root@POOJA: ~  
#!/bin/bash  
  
echo "Enter the number:"  
read num  
  
if (( num % 2 == 0 ))  
then  
    echo "The number is Even"  
else  
    echo "The number is Odd"  
fi  
|  
~
```

```
root@POOJA: ~  
root@POOJA:~# vim script3.sh  
root@POOJA:~# ./script3.sh  
-bash: ./script3.sh: Permission denied  
root@POOJA:~# chmod 755 script3.sh  
root@POOJA:~# ./script3.sh  
Enter the number:  
4  
The number is Even  
root@POOJA:~# vim script3.sh  
root@POOJA:~# |
```

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

```
root@POOJA: ~  
#!/bin/bash  
for i in {1..5}  
do  
    echo $i  
done  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

```
root@P00JA:~# vim script4.sh
root@P00JA:~# chmod 755 script4.sh
root@P00JA:~# ./script4.sh
1
2
3
4
5
root@P00JA:~# vim script4.sh
root@P00JA:~# |
```

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

Question

```
#!/bin/bash

i=1

while [ $i -le 5 ]
do
    echo $i
    i=$((i + 1))
done
|
~
~
~
```

```
root@P00JA:~# vim script5.sh
root@P00JA:~# ./script5.sh
1
2
3
4
5
root@P00JA:~# |
```

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"

```
#!/bin/bash

if [ -f "file.txt" ]
then
    echo "File exists"
else
    echo "File does not exist"
fi

~
~
~
~
```

```
root@P00JA:~# vim script6.sh
root@P00JA:~# chmod 755 script6.sh
root@P00JA:~# ./script6.sh
File does not exist
root@P00JA:~# vim script6.sh
root@P00JA:~# |
```

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.


```
root@POOJA: ~
#!/bin/bash

echo "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
|
~
~
~
```

```
root@POOJA:~# ./script7.sh
Enter a number:
6
The number is not greater than 10.
root@POOJA:~# vim script7.sh
root@POOJA:~# ./script7.sh
Enter a number:
8
The number is not greater than 10.
root@POOJA:~# |
```

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 representing the multiplication result for that number

```
root@P00JA: ~  
#!/bin/bash  
  
echo "Multiplication Table:"  
  
for i in {1..5}  
do  
    for j in {1..5}  
    do  
        result=$((i * j))  
        printf "%-4d" $result  
    done  
    echo "" # Move to the next line after each row  
done  
~
```

```
The number is not greater than 10.  
root@P00JA:~# vim script9.sh  
root@P00JA:~# chmod 755 script9.sh  
root@P00JA:~# ./script9.sh  
Multiplication Table:  
1  2  3  4  5  
2  4  6  8  10  
3  6  9  12 15  
4  8  12 16 20  
5  10 15 20 25  
root@P00JA:~# vim script9.sh  
root@P00JA:~# |
```