Assignment 7

1. Write a shell script to display your LOGIN NAME and HOME directory.
code:

```
#!/bin/bash

echo "Login Name : $USER"
echo "Home Directory : $HOME"
```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb1.sh
Login Name : kedar
Home Directory : /home/kedar
```

2. Write a shell script to display menu like “1. Date, 2. Cal, 3. Ls, 4. Pwd, 5. Exit” and execute the commands depending on user choice.
CODE :

```
#!/bin/bash

while true
do
    echo "----------------menu-----------"
    echo "1. Date"
    echo "2. Calender"
    echo "3. List files (ls)"
    echo "4. Present Working Directory"
    echo "5. Exit"
    echo "------------------------------"
    read -p "Enter your choice (1-5) " ch

    case $ch in
        1) date ;;
        2) cal ;;
        3) ls ;;
        4) pwd ;;
        5) echo "Exiting"; break;;
        *) echo "Invalid choice!" ;;
    esac
done

```

Output :

```
------------------------------
Enter your choice (1-5) 3
demo1.sh  prb1.sh  prb2.sh
---------------menu----------
1. Date
2. Calender
3. List files (ls)
4. Present Working Directory
5. Exit
------------------------------
Enter your choice (1-5) 4
/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7
---------------menu----------
1. Date
2. Calender
3. List files (ls)
4. Present Working Directory
5. Exit
------------------------------
Enter your choice (1-5) 5
Exiting
```

3. Write a shell script to accept the name from the user and check whether user entered name is file or directory. If name is file display its size and if it is directory display its contents.

CODE:

```
#!/bin/bash
read -p "Enter name: " name

if [ -f "$name" ]; then
    echo "$name is a file"
echo "Size : $(stat -c%s "$name") bytes"
elif [ -d "$name" ]; then
    echo "$name is a Directory"
    echo "contents: "
    ls "$name"
else
    echo "$name doesn't exist"
fi

```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb3.sh
Enter name: users
users is a Directory
contents:
user1.txt  user2.txt
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb3.sh
Enter name: prb3
prb3 doesn't exist
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb3.sh
Enter name: prb2.sh
prb2.sh is a file
Size : 398 bytes
```

4. Write a shell script to determine whether a given number is prime or not

Code :

```
#!/bin/bash

read -p "Enter a number : " num

for ((i=2 ; i<=num/2 ; i++))
do
    if [ $((num%i)) -eq 0 ]; then
        echo "$num is not a prime number "
        exit
    fi
    done

echo "$num is a prime number"

```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb4.sh
Enter a number : 55
55 is not a prime number
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb4.sh
Enter a number : 19
19 is a prime number
```

5. Write a Program to find the greatest of three numbers

Code :

```
#!/bin/bash

read -p "Enter num1 : " num1
read -p "Enter num2 : " num2
read -p "Enter num3 : " num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]; then
    echo "$num1 is Largest number"
elif [ $num2 -gt $num3 ]; then
    echo "$num2 is Largest number"
else
echo "$num3 is Largest number"
fi

```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb5.sh
Enter num1 : 5
Enter num2 : 67
Enter num3 : 8
67 is Largest number
```

6. Write a Program to find whether a given year is a leap year or not

Code :

```
#!/bin/bash

read -p "Enter the year" year

if [ $((year%4)) -eq 0 ]; then
    echo "$year is leap year "
else
    echo "$year is not a leap year"
fi
```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb6.sh
Enter the year2000
2000 is leap year
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb6.sh
Enter the year2001
2001 is not a leap year
```

7. Write a Program to find whether a given number is positive or negative

Code :

```
#!/bin/bash

read -p "Enter a  number : " num

if [ $num -lt 0 ]; then
    echo "$num is a negative number"
else
    echo "$num is a Positive number"
fi
```

Output :

```
Enter a  number : 44
44 is a Positive number
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb7.sh
Enter a  number : -22
-22 is a negative number
```

8. Write a program to print the table of a given number.
Code :

```
#!/bin/bash

read -p "Enter a number : " n

for ((i=1 ; i<=10 ; i++ ))
do
    echo "$n X $i = $((n*i))"
done
```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb8.sh
Enter a number : 5
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
```

9. Write a program to find the factorial of given number.
Code :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb9.sh
Enter a number : 5
Factorial of 5 is : 120
```

Output :

```
#!/bin/bash

read -p "Enter a number : " num
ans=1
for ((i=1 ;i<=num ;i++))
do
    ans=$((i*ans))
done

echo "Factorial of $num is : $ans"
```

10. Write a program to find given number of terms in the Fibonacci series.
Code :

```
#!/bin/bash
read -p "Enter number of terms: " n

a=0
b=1

echo "Fibonacci Series: "
for ((i=0; i<n; i++))
do
   echo -n "$a "
   fn=$((a + b))
   a=$b
   b=$fn
done
echo

```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb10.sh
Enter number of terms: 7
Fibonacci Series:
0 1 1 2 3 5 8
```

11. Write a program to calculate gross salary if the DA is 40%, HRA is 20% of basic salary. Accept basic salary form user and display gross salary (Result can be floating point value).
Code :

```
#!/bin/bash

DA_PERCENT=0.40
HRA_PERCENT=0.20

echo -n  "enter the basic salary : "
read BASIC_SALARY

DA=$(echo "scale=2; $BASIC_SALARY * $DA_PERCENT" | bc)
HRA=$(echo "scale=2; $BASIC_SALARY * $HRA_PERCENT" | bc)
GROSS_SALARY=$(echo "scale=2; $BASIC_SALARY + $DA + $HRA" | bc)

echo "--- Salary Details ---"
echo "Basic Salary: $BASIC_SALARY"
echo "Dearness Allowance (DA): $DA"
echo "House Rent Allowance (HRA): $HRA"
echo "Gross Salary: $GROSS_SALARY"
echo "----------------------"
```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb11.sh
enter the basic salary : 400000
--- Salary Details ---
Basic Salary: 400000
Dearness Allowance (DA): 160000.00
House Rent Allowance (HRA): 80000.00
Gross Salary: 640000.00
---------------------
```

12. Write a shell script to accept a filename as argument and displays the last modification time if the file exists and a suitable message if it doesn't exist.

Code :

```
#!/bin/bash

echo -n "enter the file path: "
read FILE

if [ -f "$FILE" ]
then

echo "File '$FILE' exists "
modtime=$(stat -c "%y" "$FILE")
echo "Last modification time: $modtime"
else
echo "error: file '$FILE' not exist or is not a regular file "
fi
```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb12.sh
enter the file path: prb11.sh
File 'prb11.sh' exists
Last modification time: 2025-11-03 14:26:04.236318700 +0530
```

13. Write a shell script to display only hidden file of current directory.

Code :

```
#!/bin/sh
shopt -s dotglob
idden_files=$(ls -A | grep '^\.' | wc -l)
if [ "$hidden_files" -gt 0 ]; then
  echo "Hidden files and directories in the current directory:"
  for file in .*; do
    # Exclude "." (current directory) and ".." (parent directory)
    if [ "$file" != "." ] && [ "$file" != ".." ]; then
      echo "$file"
    fi
  done
else
  echo "No hidden files found in the current directory."
fi

```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb13.sh
No hidden files found in the current directory.
```

14. Write a shell script to display only executable files of current directory.

Code :

```
#!/bin/bash
found_executable=false
for item in *; do
if [ -f "$item" ] && [ -x "$item" ]; then
    echo "$item"
    found_executable=true
  fi
done
if [ "$found_executable" = false ]; then
  echo "No executable files found in the current directory."
fi

```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb14.sh
No executable files found in the current directory.
```

15. Accept the two file names from user and append the contents in reverse case of first file into second file.
Code :

```bash
#!/bin/bash

echo "Enter the name of the first file (source):"
read source_file

echo "Enter the name of the second file (destination):"
read dest_file

if [ ! -r "$source_file" ]; then
    echo "Error: Source file '$source_file' does not exist or is not readable."
    exit 1
fi

tr 'a-zA-Z' 'A-Za-z' < "$source_file" >> "$dest_file"
echo "Contents of '$source_file' have been appended to '$dest_file' with reversed case."
```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb15.sh
Enter the name of the first file (source):
a.txt
Enter the name of the second file (destination):
b.txt
Contents of 'a.txt' have been appended to 'b.txt' with reversed case.
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ cat b.txt
Bile
mY NAME IS "kEDAR"
```

16. Write a shell script to display welcome message to the user along with contents of his home directory. Ensure that shell script will execute automatically when user login to the shell. (Make entry of your shell script into .bashrc file into your home directory).
Code :
Output :

17. Print the following pattern.

```
*
* *
* * *
* * * *
* * * * *
```

Code:

```
#!/bin/bash
rows=5
for ((i=1; i<=rows; i++))
 do
 for ((j=1; j<=i; j++))
 do
    echo -n "* " # Use echo -n to print without a newline
  done
 echo
done
```

Output :

```
kedar@D3-Dhananjay-92969:/media/kedar/New Volume/11_CDAC/07_OS/assignment/assign7$ bash prb17.sh
*
* *
* * *
* * * *
* * * * *
```