Assignment shell script

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

PROGRAM:->

```
#!/bin/bash
# Display login name
echo "Login Name: $USER"
# Display home directory
echo "Home Directory: $HOME"
```

OUTPUT->

```
(base) saifaliawati@saifalis-MacBook-Air ~ % vim HOME.sh

(base) saifaliawati@saifalis-MacBook-Air ~ % bash  HOME.sh

Login Name: saifaliawati
Home Directory: /Users/saifaliawati
(base) saifaliawati@saifalis-MacBook-Air ~ %
```

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.

PROGRAM:->

```
(base) saifaliawati@saifalis-MacBook-Air ~ % vim MENU.sh
#!/bin/bash
while true; do
   # Display menu
   echo "Menu:"
   echo "1. Date"
   echo "2. Calendar"
   echo "3. List Files"
   echo "4. Print Working Directory"
   echo "5. Exit"
   # Get user input
   read -p "Enter your choice (1-5): " choice
   # Execute commands based on user choice
   case $choice in
      1)
         date
         ;;
      2)
         cal
         ;;
      3)
```

```
            ls
            ;;
    4)
            pwd
            ;;
        5)
            echo "Exiting..."
            exit 0
            ;;
        *)
            echo "Invalid choice. Please enter a number between 1 and 5."
            ;;
    esac
    # Prompt to continue or exit

    read -p "Do you want to continue? (y/n): " continue_choice
    # Exit if the user chooses 'n'

    [ "$continue_choice" != "y" ] && echo "Exiting..." && exit 0
done
```

OUTPUT->

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash MENU.sh
Menu:
1. Date
2. Calendar
3. List Files
4. Print Working Directory
5. Exit
Enter your choice (1-5): 1
Thu Dec 28 14:52:21 IST 2023
Do you want to continue? (y/n): y
Menu:
1. Date
2. Calendar
3. List Files
4. Print Working Directory
5. Exit
Enter your choice (1-5): 2
   December 2023
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
Do you want to continue? (y/n): y
Menu:
1. Date
2. Calendar
3. List Files
4. Print Working Directory
```

```
5. Exit
Enter your choice (1-5): 3
<!DOCTYPE html>.html    HOME.sh                 Pictures            avanti.save
hardlink.txt            new1                    output.txt          saif23.txt.save
AndroidStudioProjects   LAB                     Postman             cal.sh
index.html              new1.save               package-lock.json   saifi
Applications            Library                 Public              corejava        input.txt
        new2                    package.json            sample.txt
Day1 update             MENU.sh                 REACT                       data.txt
install.sh      news                    rename.txt          sample1.txt
Desktop                 Movies                  Shivam              demo1.txt               java
                        newsapp                 s                   shortcut.txt
Documents               Music                   Software            ex                      java.java
        node_modules            sa                      three
Downloads               NEW                     anaconda3           example.txt             k.txt
                        one                     saif                try
HOME.TXT                OneDrive        as                          file1.txt       my_file.txt
        opt                     saif.css        user
Do you want to continue? (y/n): y
Menu:
1. Date
2. Calendar
3. List Files
4. Print Working Directory
5. Exit
Enter your choice (1-5): 4
/Users/saifaliawati
Do you want to continue? (y/n): y
Menu:
1. Date
2. Calendar
3. List Files
4. Print 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.

PROGRAM:

```
#!/bin/bash
echo "Enter a file or directory name:"
read name
if [ -f "$name" ]; then
   echo "It is a file. Its size is:"
   du -sh "$name"
elif [ -d "$name" ]; then
   echo "It is a directory. Its contents are:"
```

```
    ls "$name"
else
    echo "It is neither a file nor a directory."
fi
```

OUTPUT:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  file.sh
Enter a file or directory name:
example.txt
It is a file. Its size is:
4.0K        example.txt
```

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

PROGRAM:-

```
!/bin/bash
echo "Enter a number:"
read number
if [ $number -lt 2 ]; then
    echo "It is not a prime number."
else
    for ((i=2; i<=$number/2; i++)); do
        if [ $(($number % $i)) -eq 0 ]; then
            echo "It is not a prime number."
            exit
        fi
    done
    echo "It is a prime number."
fi
```

OUTPUT:-

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash PRIME.sh
Enter a number:
0
It is not a prime number.
(base) saifaliawati@saifalis-MacBook-Air ~ % bash PRIME.sh
Enter a number:
2
It is a prime number.
(base) saifaliawati@saifalis-MacBook-Air ~ % bash PRIME.sh
Enter a number:
3
It is a prime number.
(base) saifaliawati@saifalis-MacBook-Air ~ % bash PRIME.sh
Enter a number:
4
```

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

PROGRAM:-

```
#!/bin/bash
echo "Enter the first number:"
read number1
echo "Enter the second number:"
read number2
echo "Enter the third number:"
read number3
```

```
if [ $number1 -gt $number2 ] && [ $number1 -gt $number3 ]; then
    echo "The greatest number is $number1."
elif [ $number2 -gt $number1 ] && [ $number2 -gt $number3 ]; then
    echo "The greatest number is $number2."
else
    echo "The greatest number is $number3."
fi
```

OUTPUT:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  GRESTEST.sh
Enter the first number:
12
Enter the second number:
14
Enter the third number:
1
The greatest number is 14.
```

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

PROGRAM:-

```
#!/bin/bash
echo "Enter a year:"
read year
if [ $((year % 4)) -eq 0 ]; then
    if [ $((year % 100)) -eq 0 ]; then
        if [ $((year % 400)) -eq 0 ]; then
            echo "$year is a leap year."
        else
            echo "$year is not a leap year."
        fi
    else
        echo "$year is a leap year."
    fi
else
    echo "$year is not a leap year."
fi
```

OUTPUT:-

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  LEAP.sh
Enter a year:
2012
2012 is a leap year.
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  LEAP.sh
Enter a year:
2013
2013 is not a leap year.
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  LEAP.sh
Enter a year:
2016
2016 is a leap year.
```

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

PROGRAM:-

```
#!/bin/bash
echo "Enter a number:"
read number
if [ $number -gt 0 ]; then
   echo "$number is a positive number."
elif [ $number -lt 0 ]; then
   echo "$number is a negative number."
else
   echo "$number is zero."
fi
```

OUTPUT:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  pos.sh
Enter a number:
0
0 is zero.
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  pos.sh
Enter a number:
-1
-1 is a negative number.
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  pos.sh
Enter a number:
4
4 is a positive number.
```

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

PROGRAM:-

```
#!/bin/bash
echo "Enter a number:"
read number
echo "Table for $number:"
for i in {1..10}; do
   result=$((number * i))
   echo "$number * $i = $result"
Done
```

OUTPUT:-

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash TABLE.sh
Enter a number:
9
Table for 9:
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
```

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

```
#!/bin/bash

function factorial {
```

```
    if [ $1 -eq 0 ]
    then
        echo 1
    else
        local result=$(( $1 * $(factorial $(( $1 - 1 ))) ))
        echo $result
    fi
}

read -p "Enter a number: " number
echo "Factorial of $number is $(factorial $number)"
```

OUTPUT:

Enter a number: 4

Factorial of 4 is 24

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

PROGRAM:

```
#!/bin/bash

read -p "Enter the number of terms: " n

a=0
b=1

echo "Fibonacci series up to $n terms:"

for i in $(seq 1 $n)
do
    echo -n "$a "
    fn=$((a + b))
    a=$b
    b=$fn
done
echo # newline at the end
```

OUTPUT:-

(base) saifaliawati@saifalis-MacBook-Air ~ % bash FIB.sh
Enter the number of terms: 234
Fibonacci series up to 234 terms:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073 4807526976 7778742049 12586269025 20365011074 32951280099 53316291173 86267571272 139583862445 225851433717 365435296162 591286729879 956722026041 1548008755920 2504730781961 4052739537881 6557470319842 10610209857723 17167680177565 27777890035288 44945570212853 72723460248141 117669030460994 190392490709135 308061521170129 498454011879264 806515533049393 1304969544928657 2111485077978050 3416454622906707 5527939700884757 8944394323791464 14472334024676221 23416728348467685 37889062373143906 61305790721611591 99194853094755497 160500643816367088 259695496911122585 420196140727489673 679891637638612258 1100087778366101931 1779979416004714189 2880067194370816120 4660046610375530309 7540113804746346429 -6246583658587674878 1293530146158671551 -4953053512429003327 -3659523366270331776 -8612576878699335103 6174643828739884737 -2437933049959450366 3736710778780434371 1298777728820984005 5035488507601418376 6334266236422402381 -7076989329685730859 -742723093263328478 -7819712422949059337 -8562435516212387815 2064596134548104464 -6497839381664283351 -4433243247116178887 7515661444929089378 3082418197812910491 -7848664430967551747 -4766246233154641256 5831833409587358613 1065587176432717357 6897420586020075970 7963007762452793327 -3586315725236682319 4376692037216111008 790376311979428689 5167068349195539697 5957444661174968386 -7322231063339043533 -1364786402164075147 -8687017465503118680 8394940206042357789 -292077259460760891 8102862946581596898 7810785687120836007 -2533095440007118711 5277690247113717296 2744594807106598585 8022285054220315881 -7679864212382637150 342420841837678731 -7337443370544958419

-6995022528707279688 4114278174457313509 -2880744354249966179 1233533820207347330
-1647210534042618849 -413676713835271519 -2060887247877890368 -2474563961713161887
-4535451209591052255 -7010015171304214142 6901277692814285219 -108737478489928923
6792540214324356296 6683802735834427373 -4970401123550767947 1713401612283659426
-3256999511267108521 -1543597898983449095 -4800597410250557616 -6344195309234006711
7301951354224987289 957756044990980578 8259707399215967867 9217463444206948445
-969573230286635304 8247890213920313141 7278316983633677837 -2920536876155560638
4357780107478117199 1437243231322556561 5795023338800673760 7232266570123230321
-5419454164785647535 1812812405337582786 -3606641759448064749 -1793829354110481963
-5400471113558546712 -7194300467669028675 5851972492481976229 -1342327975187052446
4509644517294923783 3167316542107871337 7676961059402795120 -7602466472198885159
74494587203909961 -7527971884994975198 -7453477297791065237 3465294890923511181
-3988182406867554056 -522887515944042875 -4511069922811596931 -5033957438755639806
8901716712142314879 3867759273386675073 -5677268088180561664 -1809508814793886591
-7486776902974448255 9150458355941216770 1663681452966768515 -7632604264801566331
-5968922811834797816 4845216997073187469 -1123705814761610347 3721511182311577122
2597805367549966775 6319316549861543897 8917121917411510672 -3210305606436497047
5706816310975013625 2496510704538516578 8203327015513530203 -7746906353657504835
456420661856025368 -7290485691801479467 -6834065029945454099 4322193351962618050
-2511871677982836046 1810321673979782001 -701550004003054048 1108771669976727953
407221665973673905 1515993335950401858 1923215001924075763 3439208337874477621
5362423339798553384 8801631677673031005 -4282689056237967227 4518942621435063778
236253565197096551 4755196186632160329 4991449751829256880 -8700098135248134407
-3708648383418877527 6037997555042539682 2329349171623662155 8367346726666201837

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).

PROGRAM:-

```
#!/bin/bash
read -p "Enter basic salary: " basic_salary
da=$(echo "scale=2; $basic_salary * 0.4" | bc)
hra=$(echo "scale=2; $basic_salary * 0.2" | bc)
gross_salary=$(echo "scale=2; $basic_salary + $da + $hra" | bc)
echo "Gross Salary: $gross_salary"
```

Output:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  sal.sh
Enter basic salary: 10000
Gross Salary: 16000.0
```

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.

PROGRAM:

```
fi#!/bin/bash
if [ -z "$1" ]; then
```

```
    echo "Usage: $0 filename"
    exit 1
fi
filename="$1"
if [ -e "$filename" ]; then
    last_modified_time=$(stat -c %y "$filename")
    echo "Last modified: $last_modified_time"
else
    echo "File does not exist"
fi
```

OUTPUT:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  filepath1.sh
Usage: filepath1.sh filename
```

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

PROGRAM:

```
#!/bin/bash
for file in .*; do
    if [ -f "$file" ]; then
        echo "$file"
    fi
done
```

OUTPUT:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  hidden.sh
.CFUserTextEncoding
.DS_Store
.bash_profile
.emulator_console_auth_token
.gitconfig
.lesshst
.my.cnf
.mysql_history
.npmrc
.sadfg
.tcshrc
.viminfo
.vimrc
.xonshrc
.zprofile
.zsh_history
.zshrc
```

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

PROGRAM:

```
#!/bin/bash
for file in *; do
    if [ -x "$file" ]; then
        echo "$file"
    fi
done
~
```

OUTPUT:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  exe.sh
AndroidStudioProjects
Applications
Day1 update
Desktop
```

```
Documents
Downloads
LAB
Library
Movies
Music
NEW
OneDrive
Pictures
Postman
Public
REACT
Shivam
Software
anaconda3
corejava
ex
example.txt
new1
news
newsapp
node_modules
one
opt
saifi
three
user
```

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

PROGRAM:

```
#!/bin/bash
echo "Enter the first file name:"
read file1
echo "Enter the second file name:"
read file2
# Append the contents in reverse case of first file into second file
tac "$file1" >> "$file2"
```

OUTPUT:-

```
(base) saifaliawati@saifalis-MacBook-Air ~ % cat abc.txt
svhxb
svhbjn
svhbjn
dxvcbn
dvhbn
sghjk
sghj
shvj
sghj
aghs
ghj
saiu
hji
fgh
fgh
dffg
saef
sej
saiy
Sa
```

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).

PROGRAM:-

```
#!/bin/bash
# welcome_script.sh
# Display welcome message
echo "Welcome to the shell, $USER!"
# Display contents of the home directory
echo "Contents of your home directory:"
ls -l ~
```

OUTPUT:-

```
Last login: Thu Dec 28 14:25:16 on ttys003
bash: /Users/saifaliawati/welcome_script.sh: No such file or directory
(base) saifaliawati@saifalis-MacBook-Air ~ %
```

17.Print the following pattern.

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

PROGRAM:

```
#!/bin/bash
echo "Enter the number of lines to print:"
read n
for ((i=1; i<=n; i++))
do
 for ((j=1; j<=i; j++))
 do
    echo -n "* "
 done
 echo
done
```

OUTPUT:

```
(base) saifaliawati@saifalis-MacBook-Air ~ % bash  pat.sh
Enter the number of lines to print:
5
*
* *
* * *
* * * *
* * * * *
```