```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ echo "LOGIN NAME:
$USER"
LOGIN NAME: sunbeam
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ echo "HOME directory:
$HOME"
HOME directory: /home/sunbeam
```

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash menu.sh
Menu:
1. Date
2. Calendar
3. List files (ls)
4. Print working directory (pwd)
5. Exit
Enter your choice (1-5): 1
Current Date:
Wednesday 27 December 2023 05:58:56 PM IST
Do you want to continue? (y/n): y
Menu:
1. Date
2. Calendar
3. List files (ls)
4. Print working directory (pwd)
5. Exit
Enter your choice (1-5): 2
Calendar:
   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 (ls)
4. Print working directory (pwd)
5. Exit
Enter your choice (1-5): 3
List of files:
'assgn 1.pdf'      demo.sh     lsout.txt   name.txt         numbers.txt1   README.md    scripts
sorted.txt          Sunbeam.txt
 Assignment2.pdf   first.txt   menu.sh              numbers.txt   one            repeat.txt   sh.txt.save
sunbeam.txt      taste.txt
Do you want to continue? (y/n): y
Menu:
```

```
1. Date
2. Calendar
3. List files (ls)
4. Print working directory (pwd)
5. Exit
Enter your choice (1-5): 4
Current Directory:
/home/sunbeam/Documents/OP SYS practice
Do you want to continue? (y/n): y
Menu:
1. Date
2. Calendar
3. List files (ls)
4. Print working directory (pwd)
5. Exit
Enter your choice (1-5): 5
Exiting the script. Goodbye!
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$
```

```
#!/bin/bash

while true; do
    # Display menu
    echo "Menu:"
    echo "1. Date"
    echo "2. Calendar"
    echo "3. List files (ls)"
    echo "4. Print working directory (pwd)"
    echo "5. Exit"

    # Get user input
    read -p "Enter your choice (1-5): " choice

    # Execute commands based on user choice
    case $choice in
        1)
            echo "Current Date:"
            date
            ;;
        2)
            echo "Calendar:"
            cal
            ;;
        3)
            echo "List of files:"
            ls
            ;;
        4)
            echo "Current Directory:"
            pwd
            ;;
        5)
            echo "Exiting the script. Goodbye!"
            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

    if [ "$continue_choice" != "y" ]; then
        echo "Exiting the script. Goodbye!"
        exit 0
    fi
done

```

Que3

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ vim AssignQue3.sh
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue3.sh
Enter a file or directory name: sunbeam.txt
sunbeam.txt is a file.
Size of sunbeam.txt: 4.0K
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue3.sh
Enter a file or directory name: one
one is a directory.
Contents of one:
total 4
-rw-rw-r-- 1 sunbeam sunbeam    0 Dec 26 21:16 111.txt
-rw-rw-r-- 1 sunbeam sunbeam    0 Dec 26 21:16 11.txt
-rw-rw-r-- 1 sunbeam sunbeam    0 Dec 26 21:16 1.txt
drwxrwxr-x 3 sunbeam sunbeam 4096 Dec 26 21:32 two
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$
```

# code

```
#!/bin/bash

# Prompt the user for a name
read -p "Enter a file or directory name: " name

# Check if the entered name is a file
if [ -f "$name" ]; then
    echo "$name is a file."
    # Display the size of the file
    size=$(du -h "$name" | cut -f1)
    echo "Size of $name: $size"
# Check if the entered name is a directory
elif [ -d "$name" ]; then
```

```
    echo "$name is a directory."
    # Display the contents of the directory
    echo "Contents of $name:"
    ls -l "$name"
else
    echo "Invalid name or the specified file/directory does not exist."
fi
```

que 4

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ vim AssignQue4.sh
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue4.sh
Enter a number: 2
2 is a prime number.
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue4.sh
Enter a number: 4
4 is not a prime number.
```

```
#!/bin/bash

# Function to check if a number is prime
is_prime() {
    num=$1
    # 0 and 1 are not prime numbers
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ vim AssignQue4.sh
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue4.sh
Enter a number: 2
2 is a prime number.
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue4.sh
Enter a number: 4
4 is not a prime number.
    if [ $num -lt 2 ]; then
        return 1
    fi
    # Check for factors up to the square root of the number
    for ((i=2; i*i<=num; i++)); do
        if [ $((num % i)) -eq 0 ]; then
            return 1  # Not a prime number
        fi
    done
    return 0  # Prime number
}

# Prompt the user for a number
read -p "Enter a number: " number

# Check if the entered number is prime
is_prime $number
if [ $? -eq 0 ]; then
    echo "$number is a prime number."
else
    echo "$number is not a prime number."
fi
```

Que5

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue5.sh
Enter the first number: 23
Enter the second number: 44
Enter the third number: 99
The greatest number is: 99
```

```
#!/bin/bash

# Function to find the greatest of three numbers
find_greatest() {
    if [ $1 -gt $2 ] && [ $1 -gt $3 ]; then
        echo $1
    elif [ $2 -gt $1 ] && [ $2 -gt $3 ]; then
        echo $2
    else
        echo $3
    fi
}

# Prompt the user for three numbers
read -p "Enter the first number: " num1
read -p "Enter the second number: " num2
read -p "Enter the third number: " num3

# Call the function to find the greatest number
result=$(find_greatest $num1 $num2 $num3)

# Display the result
echo "The greatest number is: $result"
```

Que6

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ vim AssignQue6.sh
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue6.sh
Enter a year: 2012
2012 is a leap year.
sunbeam@sunbeam-Lenovo-B590:~/Documents/OP SYS practice$ bash AssignQue6.sh
Enter a year: 2023
2023 is not a leap year.
```

```
#!/bin/bash

# Function to check if a year is a leap year
is_leap_year() {
    year=$1

    # Check if the year is divisible by 4
    if [ $((year % 4)) -eq 0 ]; then
        # Check if the year is not divisible by 100 or it is divisible by 400
        if [ $((year % 100)) -ne 0 ] || [ $((year % 400)) -eq 0 ]; then
            return 0  # Leap year
        fi
    fi

    return 1  # Not a leap year
}

# Prompt the user for a year
read -p "Enter a year: " input_year

# Check if the entered year is a leap year
is_leap_year $input_year
if [ $? -eq 0 ]; then
    echo "$input_year is a leap year."
else
    echo "$input_year is not a leap year."
fi
```

Que8

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ vim table8.sh
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash table8.sh
Enter No :5
Table of 5:
5
10
15
20
25
30
35
40
45
50
```

```
echo -n "Enter No :"
read num
echo "Table of $num:"

#for i in 1 2 3 4 5 6 7 8 9 10
for  i in `seq 10`
do
res=`expr $i \* $num`
echo $res
#i=`expr $i + 1`
done
```

Que 9

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ vim fact9.sh
```

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash fact9.sh
Enter a number:
6
The factorial of 6 is: 720
```

```bash
#!/bin/bash

echo "Enter a number: "
read number

factorial=1

if [ $number -lt 0 ]; then
    echo "Factorial is not defined for negative numbers."
elif [ $number -eq 0 -o $number -eq 1 ]; then
    echo "The factorial of $number is: 1"
else
    for ((i=2; i<=number; i++)); do
        factorial=$((factorial * i))
    done
    echo "The factorial of $number is: $factorial"
fi
```

Que10

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ vim fibonacci10.sh

sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash fibonacci10.sh

Enter the number of terms in the Fibonacci series: 6

The first 6 terms of the Fibonacci series are: 0 1 1 2 3 5
```

```bash
#!/bin/bash


generate_fibonacci() {
    n=$1
    a=0
    b=1

    echo -n "The first $n terms of the Fibonacci series are: "

    for (( i=0; i<n; i++ )); do
        echo -n "$a "

        next_term=$((a + b))
        a=$b
        b=$next_term
    done

    echo ""  # Move to the next line after printing the series
}

# Input: Get the number of terms from the user
read -p "Enter the number of terms in the Fibonacci series: " number_of_term
generate_fibonacci $number_of_terms
```

Que11

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ vim que12.sh

sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash que12.sh

Enter the basic salary: 40000

The gross salary is :64000.0
```

```
#!bin/bash

calculate_gross_salary(){
basic_salary=$1

da_percentage=0.4
hra_percentage=0.2

da=$(echo "$basic_salary * $da_percentage" | bc)
hra=$(echo "$basic_salary * $hra_percentage" | bc)

gross_salary=$(echo "$basic_salary + $da + $hra" | bc)

echo "The gross salary is :"$gross_salary
}

 read -p "Enter the basic salary: " basic_salary
 calculate_gross_salary $basic_salary
```

Que13

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash  que13.sh

Hidden files in the current directory:

sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$
```

```
#!bin/bash

hidden_files=$(ls -a | grep &apos;^/.&apos;)

echo "Hidden files in the current directory:"
echo "$hidden_files"

```

Que14

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ vim que14.sh
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash que14.sh
Executable files:
menu.sh
one
scripts
```

```
#!bin/bash
executable_files=$(ls -l | grep -E &apos;^\S*x&apos; | awk &apos;{print $NF}&apos;)

echo "Executable files:"
echo "$executable_files"
```

Que 15

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash que15.sh
 Enter 1st file
file.txt
Enter 2nd file
file1.txt
shraddha lkijuyt kijuuhgfds fd illaveerhs
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$
```

```
#!/bin/bash

echo " Enter 1st file"
read file
echo "Enter 2nd file "
read file1
rev "$file" >> "$file1"
echo `cat $file1`
```

Que16

```
Welcome shraddha!!!
sunbeam@sunbeam-Lenovo-B590:~/Documents/git_data/osc/Assignments$
```

Que17

```
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ vim que17.sh
sunbeam@sunbeam-Lenovo-B590:~/Documents/OSpractice$ bash que17.sh
*
* *
* * *
* * * *
* * * * *
```

```
#!bin/bash

rows=5

for ((i=1; i<=rows; i++)); do
    for ((j=1; j<=i; j++)); do
        echo -n "* "
    done
    echo
done

```