OPERATING SYSTEMS

LABSHEET 3

Anuvind MP

AM.EN.U4AIE22010

1. Write shell scripts for the following:

a. To take your name, programme name and enrolment number as input from user and print it on the screen.

```
#!/bin/sh
echo "Q1-a"
read -p "Enter your name : " name
read -p "Enter programme name : " programme
read -p "Enter enrolment number : " enrolement
echo -e "Name : " $name
echo "Programme : " $programme
echo "Enrolement number : " $enrolement
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1.sh
Q1-a
Enter your name : Anuvind MP
Enter programme name : OS LAB3
Enter enrolment number : AM.EN.U4AIE22010
Name :  Anuvind MP
Programme :  OS LAB3
Enrolement number :  AM.EN.U4AIE22010
```

b. To find the sum, the average and the product of four integers.

```
#!/bin/sh

read -p "Enter a number : " a
read -p "Enter a number : " b
read -p "Enter a number : " c
read -p "Enter a number : " d

sum=$((a + b + c + d))
avg=$((sum / 4))
product=$((a * b * c * d))

echo "sum = $sum"
echo "Average = $avg"
echo "Product = $product"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1-b.sh
Enter a number : 4
Enter a number : 7
Enter a number : 9
Enter a number : 8
sum = 28
Average = 7
Product = 2016
```

c. Write a program to check whether a number is even or odd.

```
#!/bin/sh

read -p "Enter a number : " a
if (($a % 2 == 0))
then
    echo "$a is even"
else
    echo "$a is odd"
fi
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1b.sh
Enter a number : 45
45 is odd
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1b.sh
Enter a number : 12
12 is even
```

d. To exchange the values of two variables.

```
#!/bin/sh

read -p "Enter a number : " a
read -p "Enter a number : " b

temp=$a
a=$b
b=$temp

echo "After exchanging :"
echo "Variable 1 : $a"
echo "Variable 2 : $b"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1C.sh
Enter a number : 4
Enter a number : 7
After exchanging :
Variable 1 : 7
Variable 2 : 4
```

e. To find the lines containing a number in a file.

```
#!/bin/sh

read -p "Enter file name : " file
read -p "Enter number to search : " num

grep "$num" "$file"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1d.sh
Enter file name : demo.txt
Enter number to search : 1
Student Bob Essentials 23 PSAT 21 Maths 32 Cultural 18 English 94
Student Boby Essentials 43 PSAT 31 Maths 22 Cultural 8 English 93
Student Clara Essentials 18 PSAT 16 Maths 27 Cultural 12 English 45
Student Eve Essentials 8 PSAT 6 Maths 12 Cultural 13 English 5
```

f. To concatenate two strings and find the length of the resultant string.

```
#!/bin/sh

read -p "Enter string 1 : " str1
read -p "Enter string 2 : " str2

con="$str1$str2"
len=${#con}

echo "Concatenated string : $con"
echo "Length of the string : $len"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1e.sh
Enter string 1 : boom
Enter string 2 : BOOOOOM
Concatenated string : boomBOOOOOM
Length of the string : 11
```

g. To concatenate the contents of two files.

```
#!/bin/sh

read -p "Enter file 1 : " f1
read -p "Enter file 2 : " f2
cat "$f1" "$f2"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q1f.sh
Enter file 1 : demo.txt
Enter file 2 : demo2.txt
HEY THIS IS FILE !


GLUB TUBBUS WEPPLE
BOM BAM BADABUM BAM
TJ CREAM
GOOBIE WOOBIE
```

h. Write a shell script that would wait 5 seconds and then display the time

```
#!/bin/sh

sleep 5
echo "Time : $(date)"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q.sh
Time : Sun Dec 31 11:28:00 IST 2023
```

2. The length and breadth of a rectangle and radius of a circle are provided as user input. Write a shell script that will calculate the area and perimeter of the rectangle and the area and circumference of the circle.

Hint:- Area of Rectangle = L*B Perimeter of Rectangle = 2(L+B) Area of Circle = π.r2 Circumference of circle = 2. π.r

```
#!/bin/sh

read -p "Enter length of rectangle: " l
read -p "Enter breadth of rectangle: " b

perimeter=$((2 * (l + b)))
area=$((l * b))

echo "Rectangle Perimeter: $perimeter"
echo "Rectangle Area: $area"

read -p "Enter radius of circle: " r

circle_area=$(echo "scale=3; 3.141 * $r * $r" | bc)
circle_circum=$(echo "scale=3; 2 * 3.141 * $r" | bc)

echo "Circle Area: $circle_area"
echo "Circle Circumference: $circle_circum"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q2.sh
Enter length of rectangle: 4
Enter breadth of rectangle: 7
Rectangle Perimeter: 22
Rectangle Area: 28
Enter radius of circle: 5
Circle Area: 78.525
Circle Circumference: 31.410
```

3. Write a menu driven shell program to read two numbers and print the results of all the arithmetic operations. (+ , - , * , / , % , ++ , --)

```
#!/bin/bash
```

```bash
echo "Menu:"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
echo "5. Modulus"
echo "6. Increment"
echo "7. Decrement"
read -p "Enter the First Number: " num1
read -p "Enter the Second Number: " num2
read -p "Enter the Operation (1-7): " op
case $op in
    1)  res=$((num1 + num2)); echo "Sum: $res";;
    2)  res=$((num1 - num2)); echo "Difference: $res";;
    3)  res=$((num1 * num2)); echo "Product: $res";;
    4)  if (( $(echo "$num2 != 0" | bc -l) )); then
            res=$(echo "scale=2; $num1 / $num2" | bc)
            echo "Quotient: $res"
        else
            echo "Cannot divide by zero."
        fi;;
    5)  if (( $(echo "$num2 != 0" | bc -l) )); then
            res=$((num1 % num2))
            echo "Remainder: $res"
        else
            echo "Cannot find remainder when dividing by zero."
        fi;;
    6)  res=$((num1 + 1)); echo "Increment of $num1: $res";;
    7)  res=$((num1 - 1)); echo "Decrement of $num1: $res";;
    *)  echo "Invalid operation.";;
esac
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q3.sh
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 5
Enter the Second Number: 7
Enter the Operation (1-7): 1
Sum: 12
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q3.sh
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 8
Enter the Second Number: 2
Enter the Operation (1-7): 2
Difference: 6
```

```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 8
Enter the Second Number: 2
Enter the Operation (1-7): 3
Product: 16
```

```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 8
Enter the Second Number: 4
Enter the Operation (1-7): 4
Quotient: 2.00
```

```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 8
Enter the Second Number: 2
Enter the Operation (1-7): 5
Remainder: 0
```

```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 5
Enter the Second Number: 2
Enter the Operation (1-7): 6
Increment of 5: 6
```

```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 5
Enter the Second Number: 7
Enter the Operation (1-7): 7
Decrement of 5: 4
```

4. Write two separate shell scripts to find the factorial of a number using while statement and for statement.

```bash
#!/bin/bash

read -p "Enter a Number to find its Factorial: " n
fact=1
for ((i = 1; i <= n; i++)); do
    fact=$((fact * i))
done
echo "Factorial of $n is: $fact"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q4.sh
Enter a Number to find its Factorial: 5
Factorial of 5 is: 120
```

5. Given a file of numbers (one number per line), write a shell script that will find the lowest and highest number.

```
#!/bin/bash

file="demo2.txt"

if [ -f "$file" ]; then
    low=$(sort -n "$file" | head -n 1)
    high=$(sort -n "$file" | tail -n 1)

    echo "Lowest number: $low"
    echo "Highest number: $high"
else
    echo "File not found."
fi
```

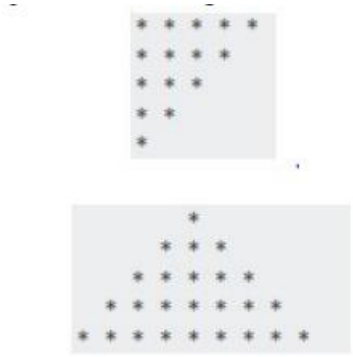
```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q5.sh
Lowest number: 2
Highest number: 874745
```

6. Write a shell program to read n numbers into an array and display the average of them.

```
#!/bin/bash

read -p "Enter length : " n
declare -a array
for ((i = 0; i < n; i++)); do
    read -p "Enter Element $((i + 1)): " array[i]
done

sum_=0
for ((i = 0; i < n; i++)); do
    sum_=$((sum_ + array[i]))
done


average=$(echo "scale=2; $sum_ / $n" | bc)
echo "Average: $average"
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q6.sh
Enter length : 5
Enter Element 1: 1
Enter Element 2: 7
Enter Element 3: 9
Enter Element 4: 6
Enter Element 5: 2
Average: 5.00
```

7. Write a shell program to print the following Patterns.



```
#!/bin/bash

echo "pattern 1 "
for ((i = 5; i >= 1; i--)); do
    for ((j = 1; j <= i; j++)); do
        echo -n "*"
    done
    echo
done

echo -e "\n"
echo "pattern 2 "
for ((i = 1; i <= 5; i++)); do
    for ((j = 5; j > i; j--)); do
        echo -n " "
    done
    for ((k = 1; k <= 2*i-1; k++)); do
        echo -n "*"
    done
    echo
done
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q7.sh
pattern 1
*****
****
***
**
*


pattern 2
    *
   ***
  *****
 *******
*********
```

8. Write a shell program to read two matrices, add them and print the output matrix.

```
#!/bin/bash

read -p "Enter the Number of Rows : " m
read -p "Enter the Number of Columns : " n
```

```bash
declare -A matrix1
declare -A matrix2

echo "Enter elements for matrix 1:"
for ((i = 0; i < m; i++)); do
    for ((j = 0; j < n; j++)); do
        read -p "Enter element at position ($((i)), $((j))) for the first matrix: "
matrix1["$i,$j"]
    done
done

echo "Enter elements for matrix 2:"
for ((i = 0; i < m; i++)); do
    for ((j = 0; j < n; j++)); do
        read -p "Enter element at position ($((i)), $((j))) for the second matrix: "
matrix2["$i,$j"]
    done
done

echo "Matrix sum:"
for ((i = 0; i < m; i++)); do
    for ((j = 0; j < n; j++)); do
        result=$((matrix1["$i,$j"] + matrix2["$i,$j"]))
        echo -n "$result "
    done
    echo
done
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q8.sh
Enter the Number of Rows : 3
Enter the Number of Columns : 3
Enter elements for matrix 1:
Enter element at position (0, 0) for the first matrix: 4
Enter element at position (0, 1) for the first matrix: 7
Enter element at position (0, 2) for the first matrix: 5
Enter element at position (1, 0) for the first matrix: 2
Enter element at position (1, 1) for the first matrix: 4
Enter element at position (1, 2) for the first matrix: 6
Enter element at position (2, 0) for the first matrix: 1
Enter element at position (2, 1) for the first matrix: 5
Enter element at position (2, 2) for the first matrix: 7
Enter elements for matrix 2:
Enter element at position (0, 0) for the second matrix: 5
Enter element at position (0, 1) for the second matrix: 7
Enter element at position (0, 2) for the second matrix: 4
Enter element at position (1, 0) for the second matrix: 1
Enter element at position (1, 1) for the second matrix: 6
Enter element at position (1, 2) for the second matrix: 8
Enter element at position (2, 0) for the second matrix: 2
Enter element at position (2, 1) for the second matrix: 1
Enter element at position (2, 2) for the second matrix: 4
Matrix sum:
9 14 9
3 10 14
3 6 11
```

9. Write a program to read a matrix and print the transpose of it.

```bash
#!/bin/bash

read -p "Enter the Number of Rows : " m
read -p "Enter the Number of Columns : " n


declare -A A

echo "Enter Elements :"
for ((i = 0; i < m; i++)); do
    for ((j = 0; j < n; j++)); do
        read -p "Enter Element at position ($((i)), $((j))): " A["$i,$j"]
    done
done

echo "Transpose of the Matrix ' :"
for ((j = 0; j < n; j++)); do
    for ((i = 0; i < m; i++)); do
        echo -n "${A["$i,$j"]} "
    done
    echo
done
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\root\lab3> bash Q9.sh
Enter the Number of Rows : 2
Enter the Number of Columns : 3
Enter Elements :
Enter Element at position (0, 0): 4
Enter Element at position (0, 1): 7
Enter Element at position (0, 2): 8
Enter Element at position (1, 0): 52
Enter Element at position (1, 1): 4
Enter Element at position (1, 2): 7
Transpose of the Matrix ' :
4 52
7 4
8 7
```