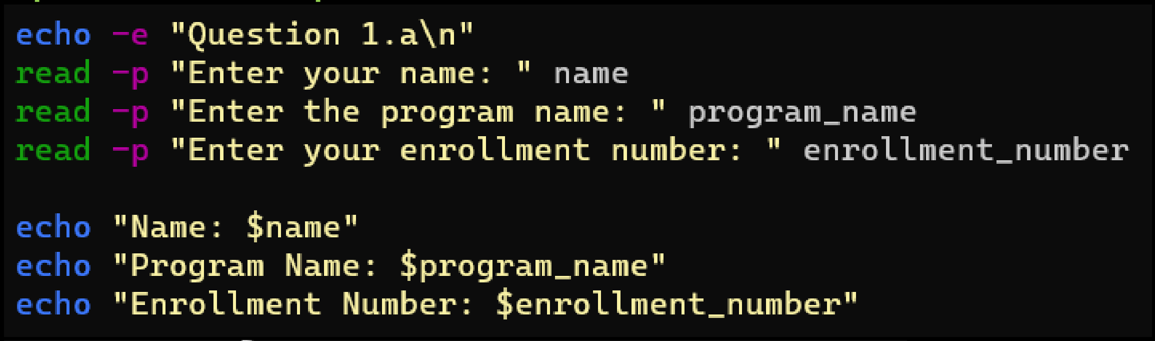
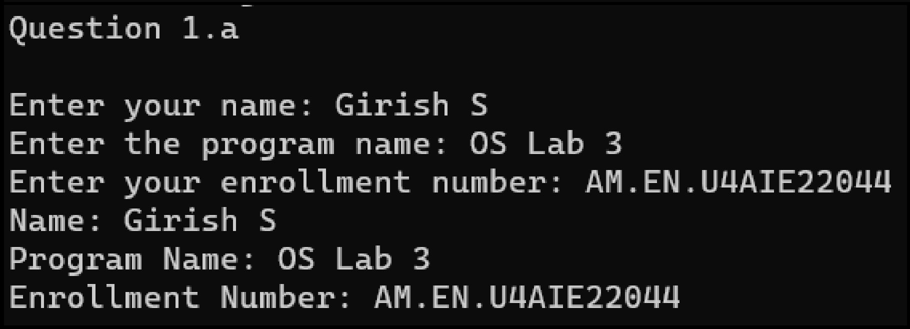
OPERATING SYSTEMS

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.

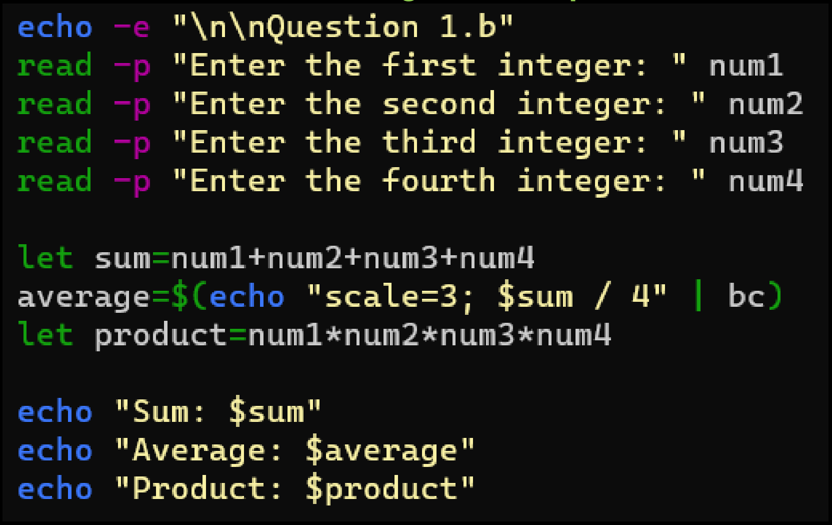


```
echo -e "Question 1.a\n"
read -p "Enter your name: " name
read -p "Enter the program name: " program_name
read -p "Enter your enrollment number: " enrollment_number

echo "Name: $name"
echo "Program Name: $program_name"
echo "Enrollment Number: $enrollment_number"
```

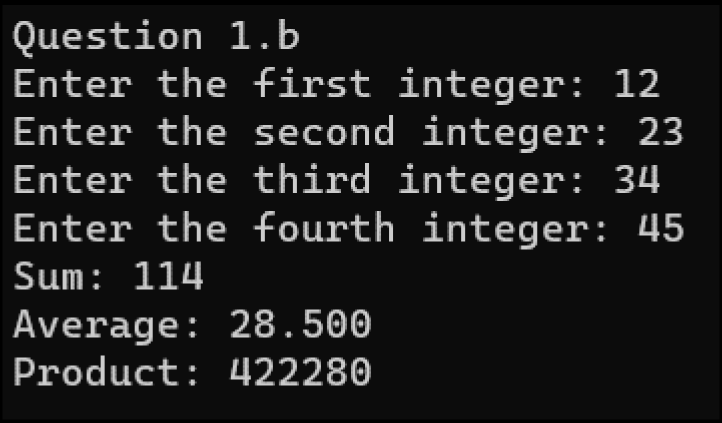


```
Question 1.a

Enter your name: Girish S
Enter the program name: OS Lab 3
Enter your enrollment number: AM.EN.U4AIE22044
Name: Girish S
Program Name: OS Lab 3
Enrollment Number: AM.EN.U4AIE22044
```

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

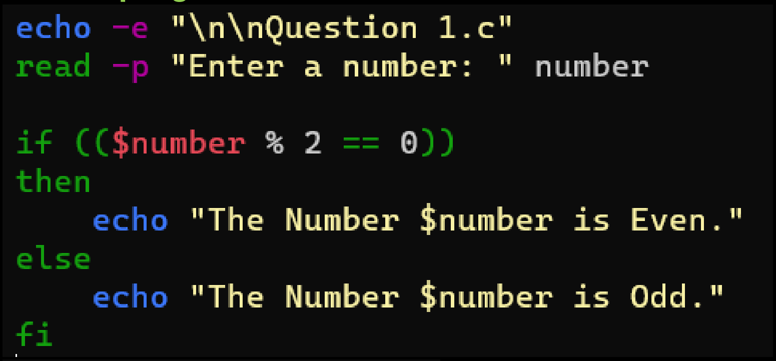


```
echo -e "\n\nQuestion 1.b"
read -p "Enter the first integer: " num1
read -p "Enter the second integer: " num2
read -p "Enter the third integer: " num3
read -p "Enter the fourth integer: " num4

let sum=num1+num2+num3+num4
average=$(echo "scale=3; $sum / 4" | bc)
let product=num1*num2*num3*num4

echo "Sum: $sum"
echo "Average: $average"
echo "Product: $product"
```

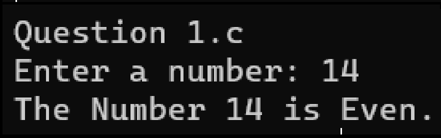


```
Question 1.b
Enter the first integer: 12
Enter the second integer: 23
Enter the third integer: 34
Enter the fourth integer: 45
Sum: 114
Average: 28.500
Product: 422280
```

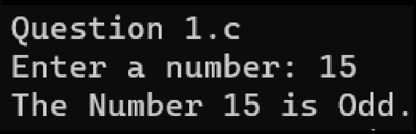
c. Write a program to check whether a number is even or odd.



```
echo -e "\n\nQuestion 1.c"
read -p "Enter a number: " number

if (($number % 2 == 0))
then
    echo "The Number $number is Even."
else
    echo "The Number $number is Odd."
fi
```

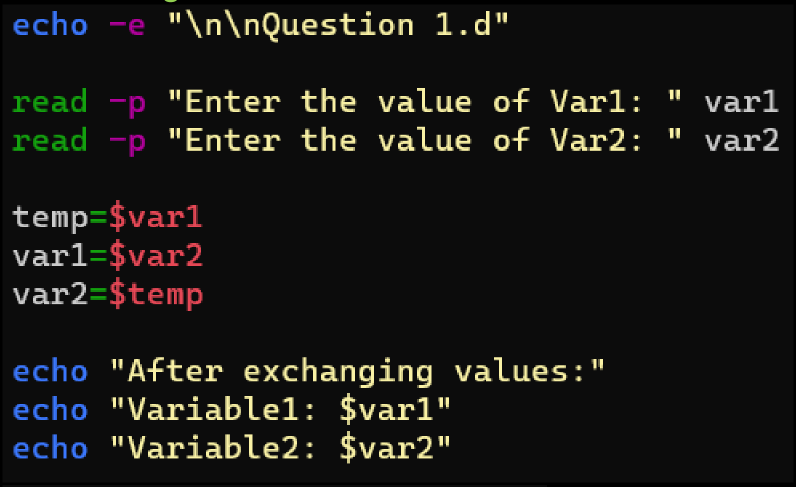


```
Question 1.c
Enter a number: 14
The Number 14 is Even.
```

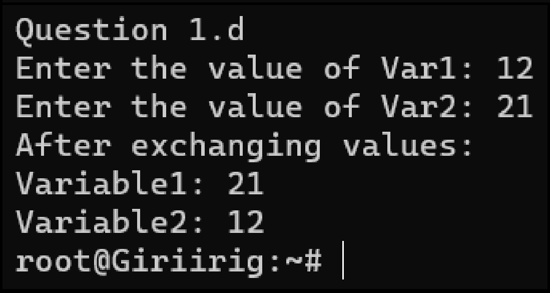


```
Question 1.c
Enter a number: 15
The Number 15 is Odd.
```

d. To exchange the values of two variables.

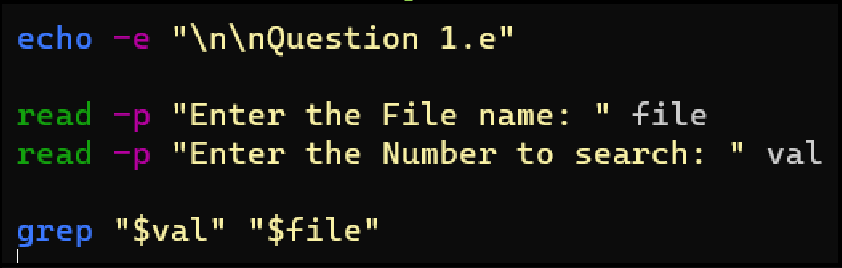


```
echo -e "\n\nQuestion 1.d"

read -p "Enter the value of Var1: " var1
read -p "Enter the value of Var2: " var2

temp=$var1
var1=$var2
var2=$temp

echo "After exchanging values:"
echo "Variable1: $var1"
echo "Variable2: $var2"
```

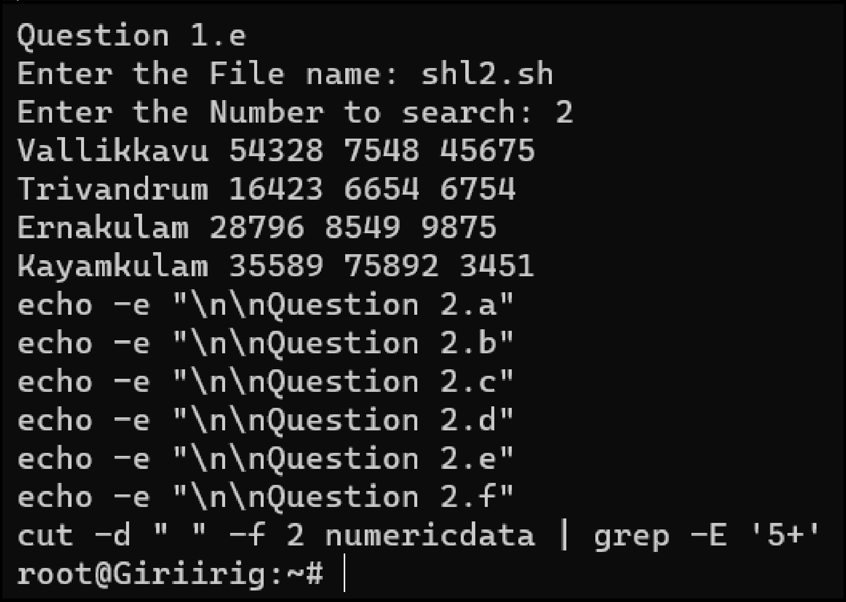


```
Question 1.d
Enter the value of Var1: 12
Enter the value of Var2: 21
After exchanging values:
Variable1: 21
Variable2: 12
root@Giriirig:~#
```

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

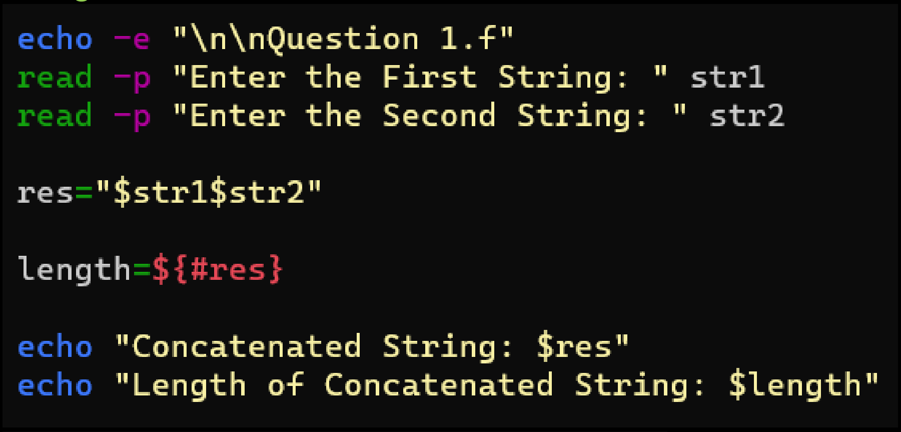


```
echo -e "\n\nQuestion 1.e"

read -p "Enter the File name: " file
read -p "Enter the Number to search: " val

grep "$val" "$file"
```

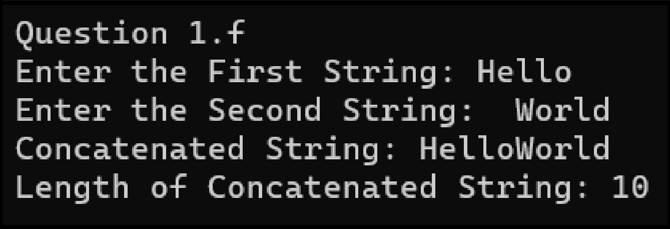


```
Question 1.e
Enter the File name: shl2.sh
Enter the Number to search: 2
Vallikkavu 54328 7548 45675
Trivandrum 16423 6654 6754
Ernakulam 28796 8549 9875
Kayamkulam 35589 75892 3451
echo -e "\n\nQuestion 2.a"
echo -e "\n\nQuestion 2.b"
echo -e "\n\nQuestion 2.c"
echo -e "\n\nQuestion 2.d"
echo -e "\n\nQuestion 2.e"
echo -e "\n\nQuestion 2.f"
cut -d " " -f 2 numericdata | grep -E '5+'
root@Giriirig:~#
```

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

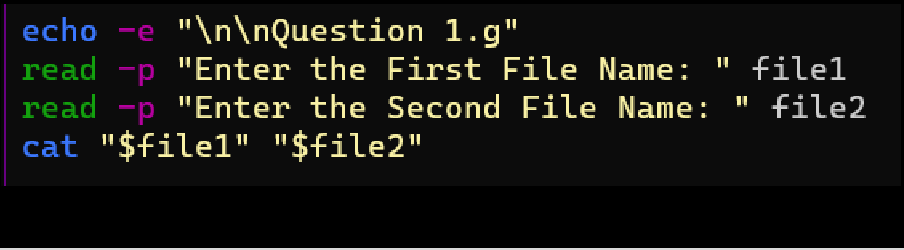


```
echo -e "\n\nQuestion 1.f"
read -p "Enter the First String: " str1
read -p "Enter the Second String: " str2

res="$str1$str2"

length=${#res}

echo "Concatenated String: $res"
echo "Length of Concatenated String: $length"
```

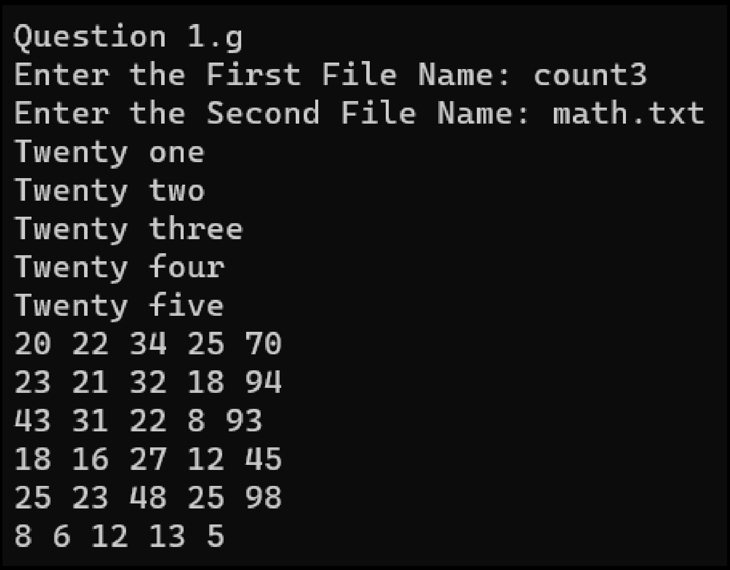


```
Question 1.f
Enter the First String: Hello
Enter the Second String:  World
Concatenated String: HelloWorld
Length of Concatenated String: 10
```

g. To concatenate the contents of two files

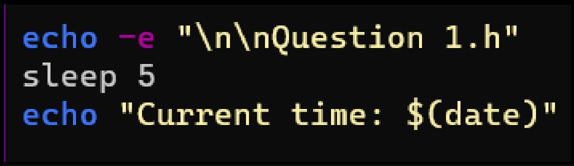


```
echo -e "\n\nQuestion 1.g"
read -p "Enter the First File Name: " file1
read -p "Enter the Second File Name: " file2
cat "$file1" "$file2"
```

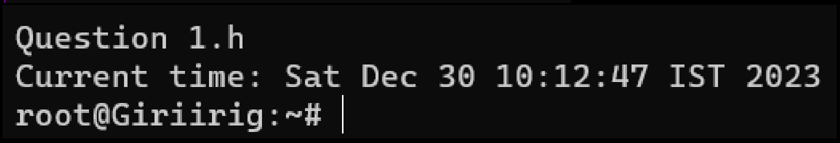


```
Question 1.g
Enter the First File Name: count3
Enter the Second File Name: math.txt
Twenty one
Twenty two
Twenty three
Twenty four
Twenty five
20 22 34 25 70
23 21 32 18 94
43 31 22 8 93
18 16 27 12 45
25 23 48 25 98
8 6 12 13 5
```

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

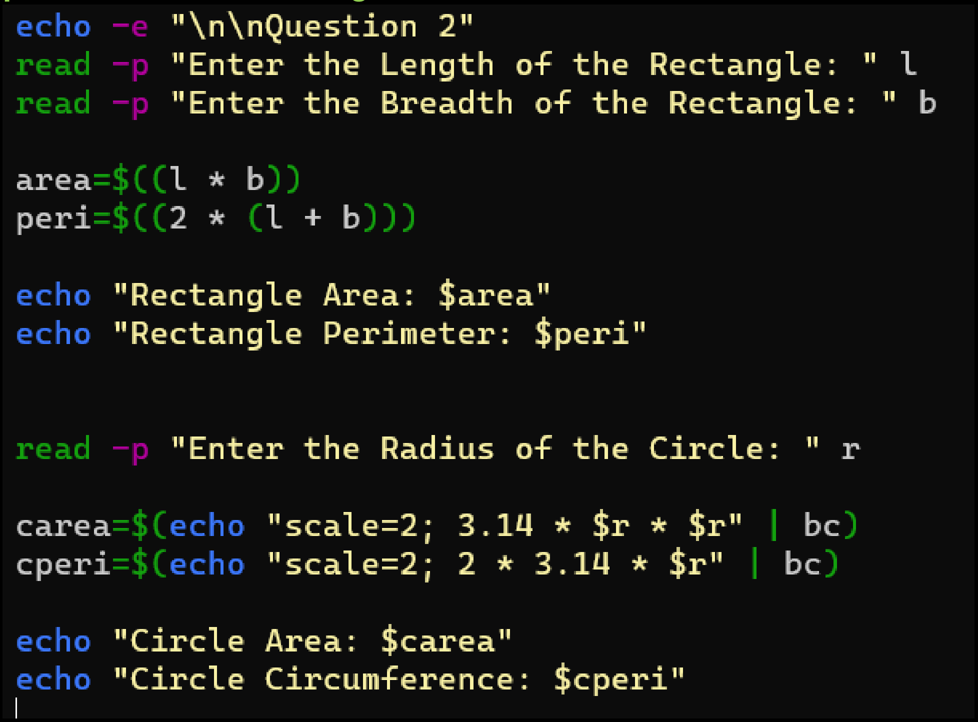


```
echo -e "\n\nQuestion 1.h"
sleep 5
echo "Current time: $(date)"
```

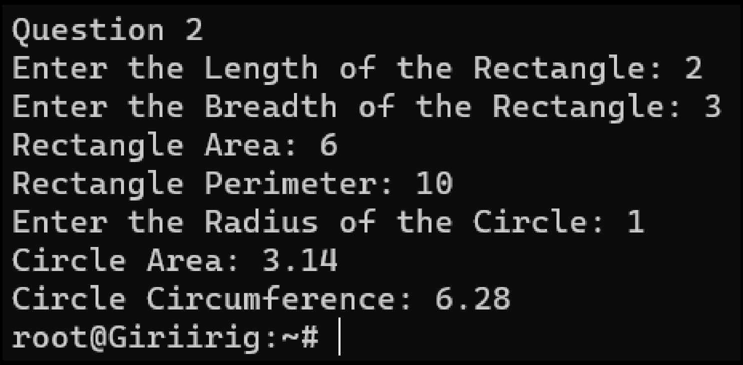


```
Question 1.h
Current time: Sat Dec 30 10:12:47 IST 2023
root@Giriirig:~#
```

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.**

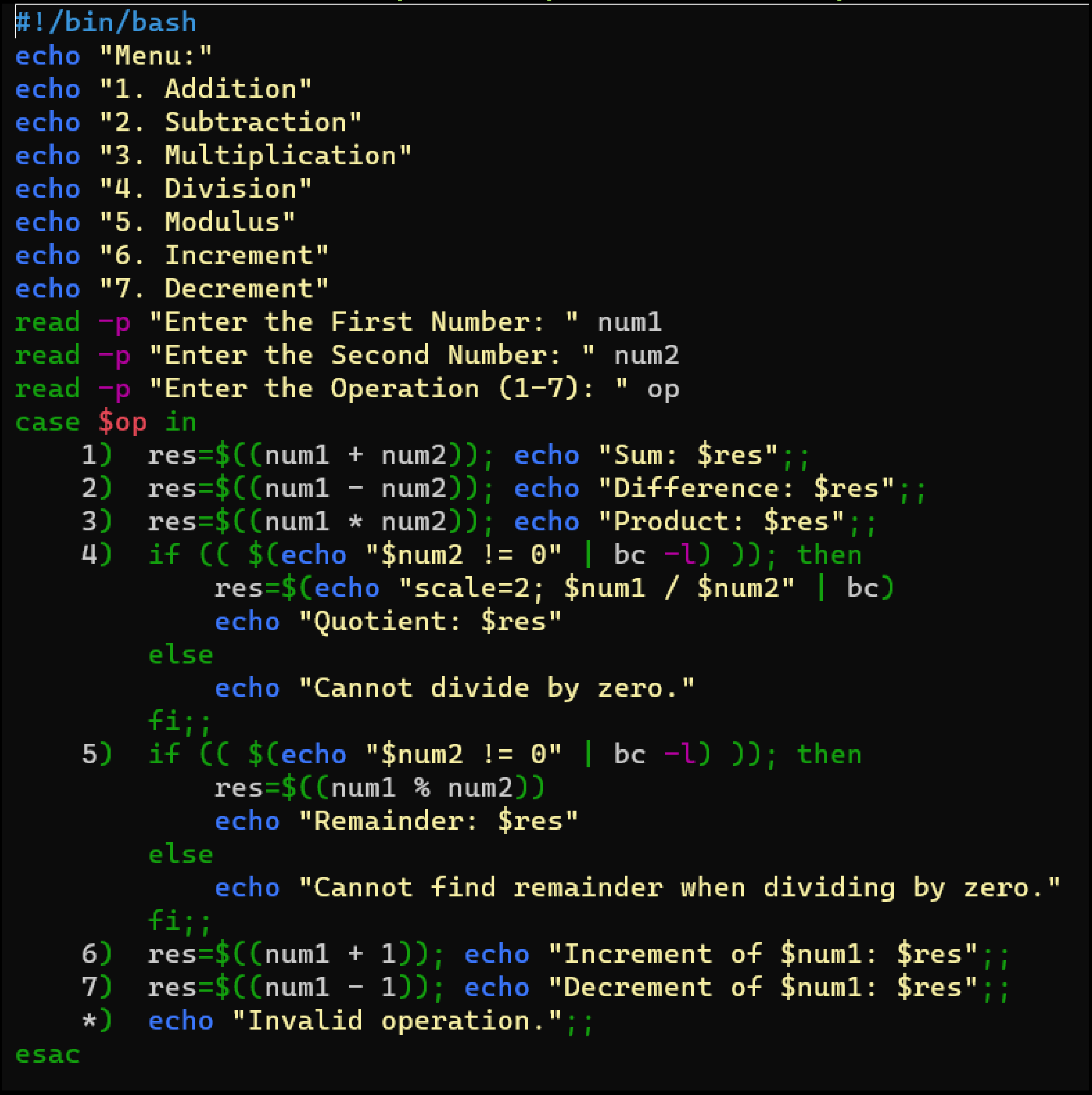


```
echo -e "\n\nQuestion 2"
read -p "Enter the Length of the Rectangle: " l
read -p "Enter the Breadth of the Rectangle: " b

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

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


read -p "Enter the Radius of the Circle: " r

carea=$(echo "scale=2; 3.14 * $r * $r" | bc)
cperi=$(echo "scale=2; 2 * 3.14 * $r" | bc)

echo "Circle Area: $carea"
echo "Circle Circumference: $cperi"
```

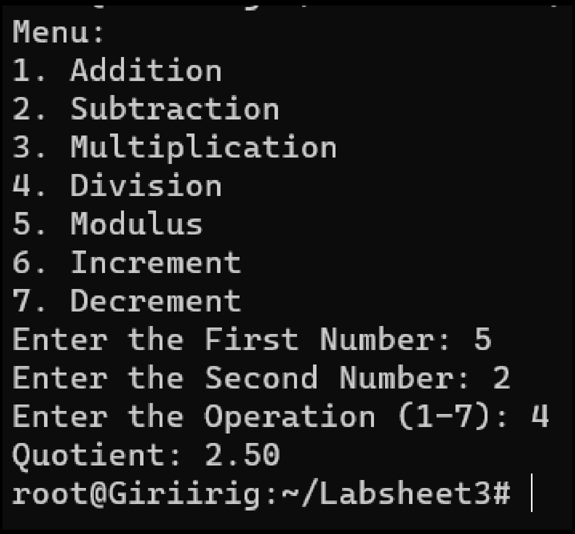


```
Question 2
Enter the Length of the Rectangle: 2
Enter the Breadth of the Rectangle: 3
Rectangle Area: 6
Rectangle Perimeter: 10
Enter the Radius of the Circle: 1
Circle Area: 3.14
Circle Circumference: 6.28
root@Giriirig:~#
```

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

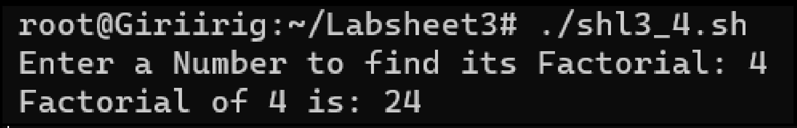


```
#!/bin/bash
echo "Menu:"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
echo "5. Modulus"
echo "6. Increment"
echo "7. Decrement"
read -p "Enter the First Number: " num1
read -p "Enter the Second Number: " num2
read -p "Enter the Operation (1-7): " op
case $op in
    1)  res=$((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
```

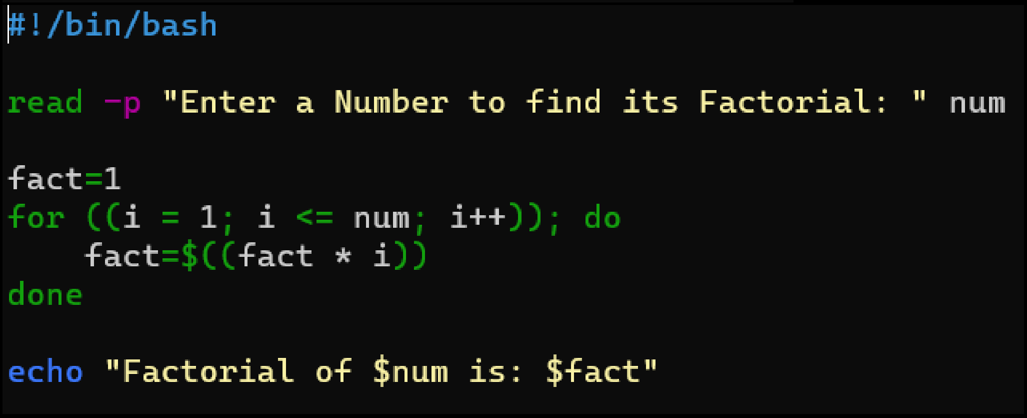


```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Increment
7. Decrement
Enter the First Number: 5
Enter the Second Number: 2
Enter the Operation (1-7): 4
Quotient: 2.50
root@Giriirig:~/Labsheet3#
```

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

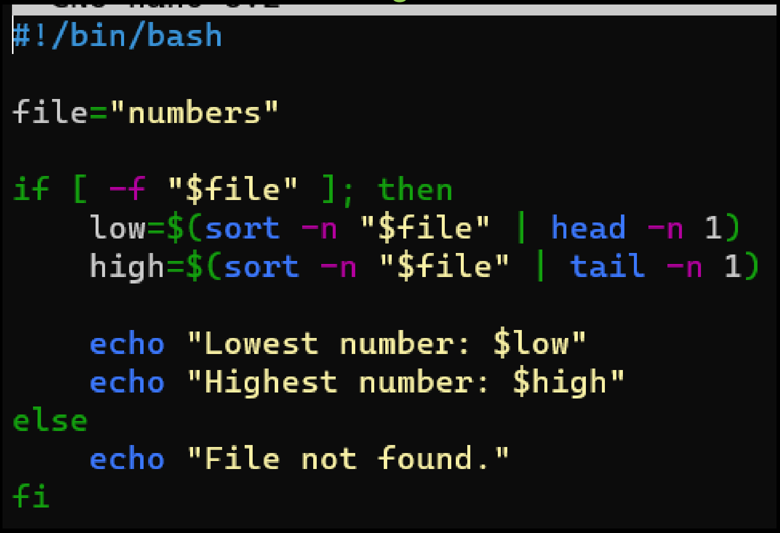


```
root@Giriirig:~/Labsheet3# ./shl3_4.sh
Enter a Number to find its Factorial: 4
Factorial of 4 is: 24
```

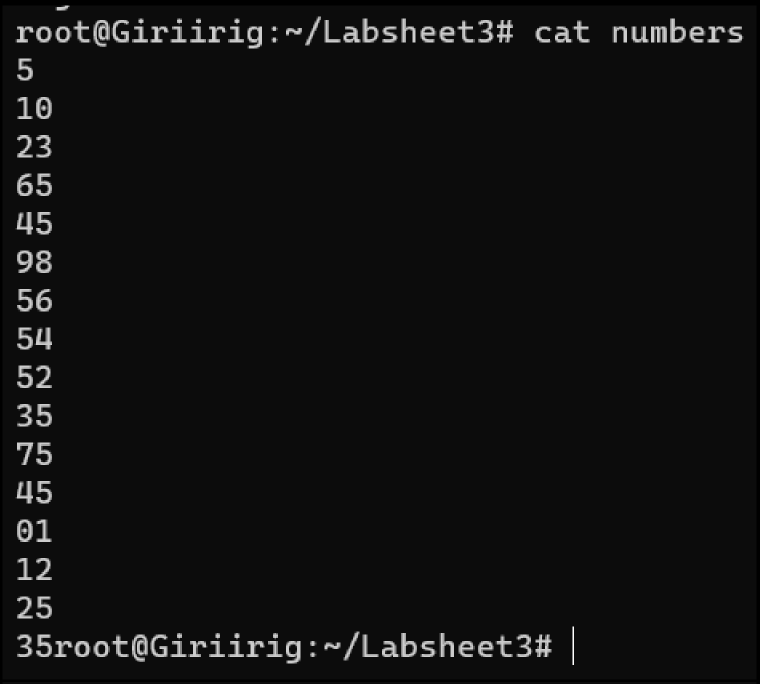


```
#!/bin/bash

read -p "Enter a Number to find its Factorial: " num

fact=1
for ((i = 1; i <= num; i++)); do
    fact=$((fact * i))
done

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

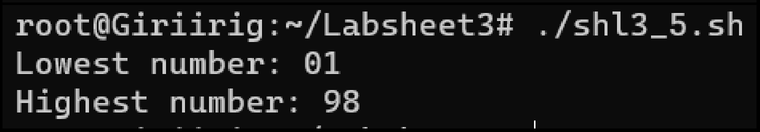
5. **Given a file of numbers (one number per line), write a shell script that will find the lowest and highest number.**



```
#!/bin/bash

file="numbers"

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
```

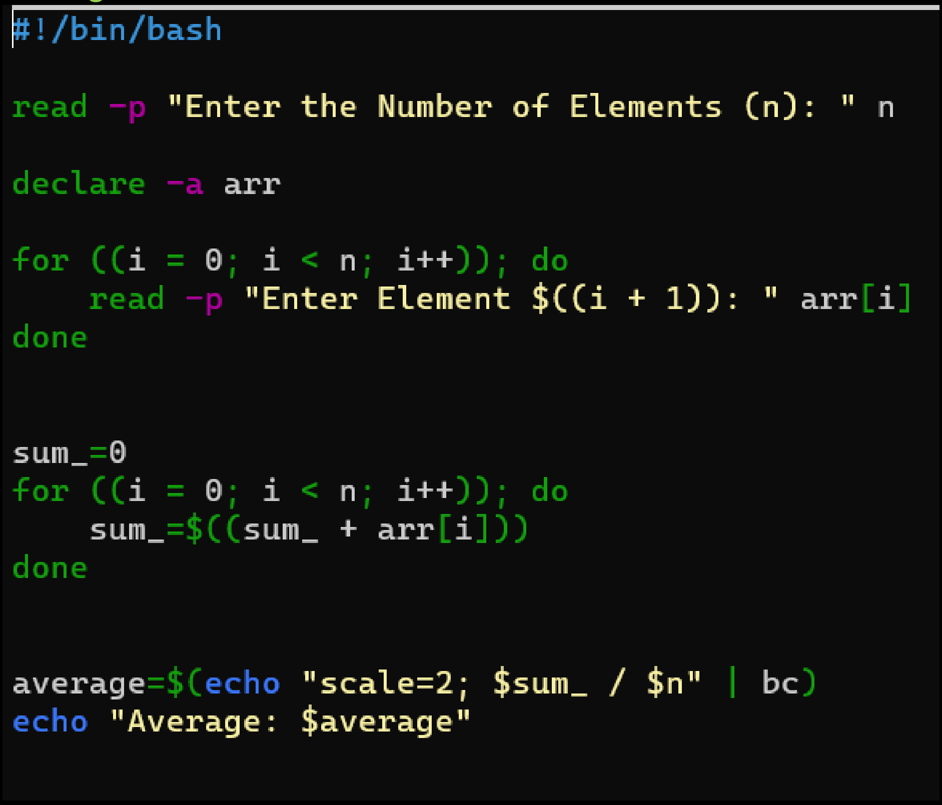


```
root@Giriirig:~/Labsheet3# cat numbers
5
10
23
65
45
98
56
54
52
35
75
45
01
12
25
35root@Giriirig:~/Labsheet3#
```

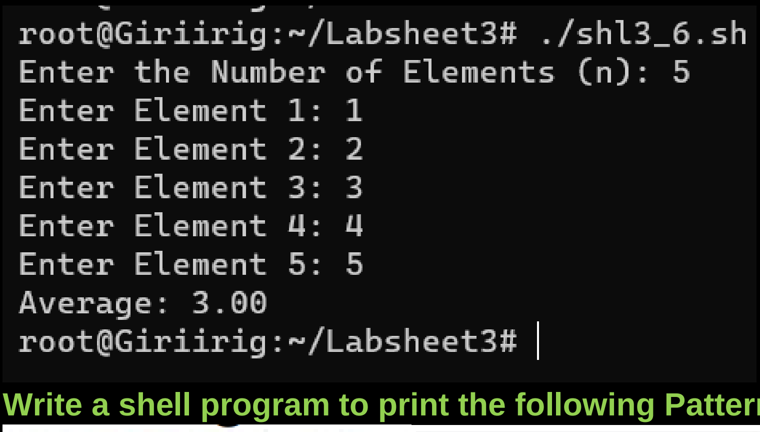


```
root@Giriirig:~/Labsheet3# ./shl3_5.sh
Lowest number: 01
Highest number: 98
```

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

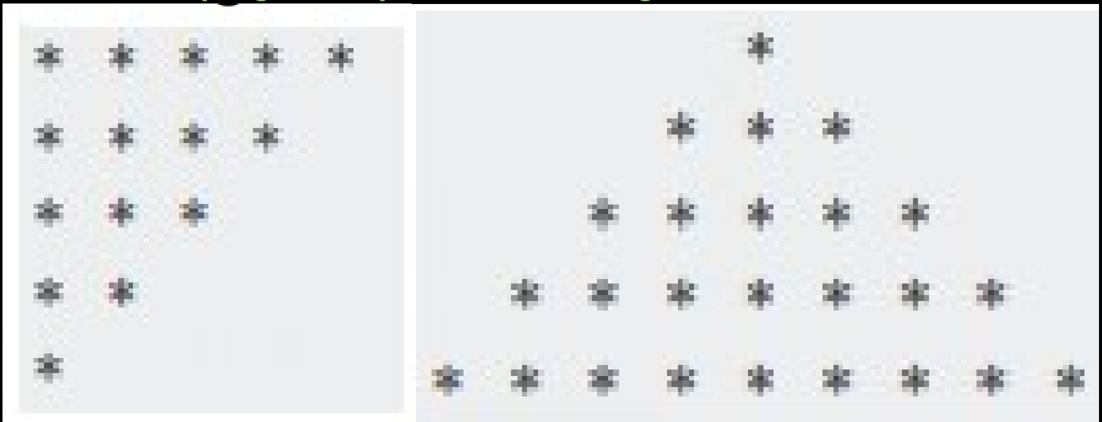


```
#!/bin/bash

read -p "Enter the Number of Elements (n): " n

declare -a arr

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


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


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

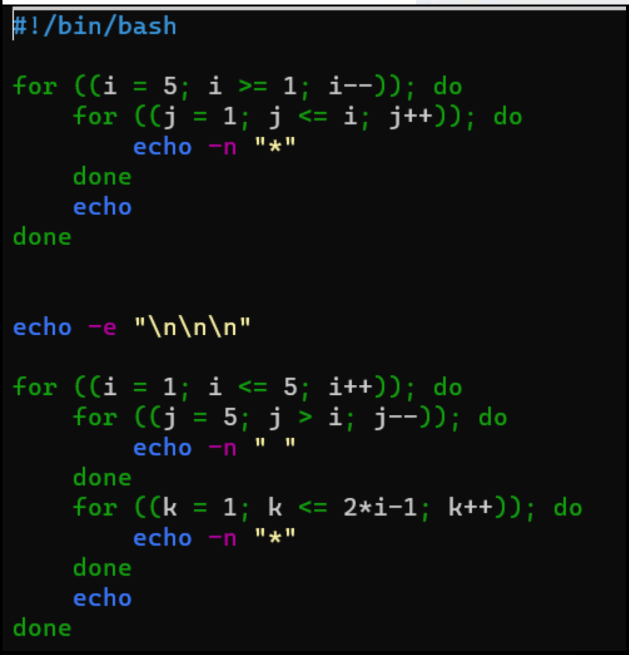


```
root@Giriirig:~/Labsheet3# ./shl3_6.sh
Enter the Number of Elements (n): 5
Enter Element 1: 1
Enter Element 2: 2
Enter Element 3: 3
Enter Element 4: 4
Enter Element 5: 5
Average: 3.00
root@Giriirig:~/Labsheet3# 
```

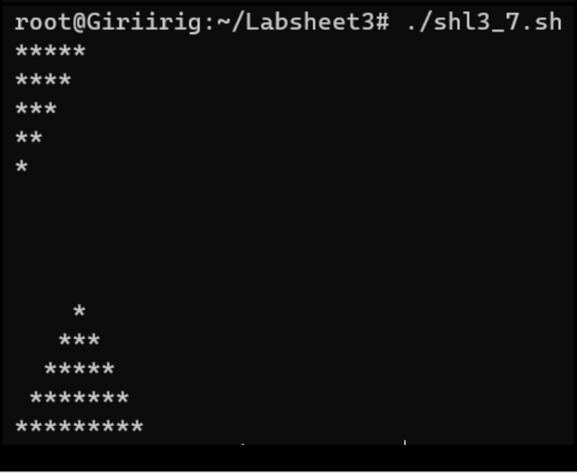
7. **Write a shell program to print the following Patterns.**



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

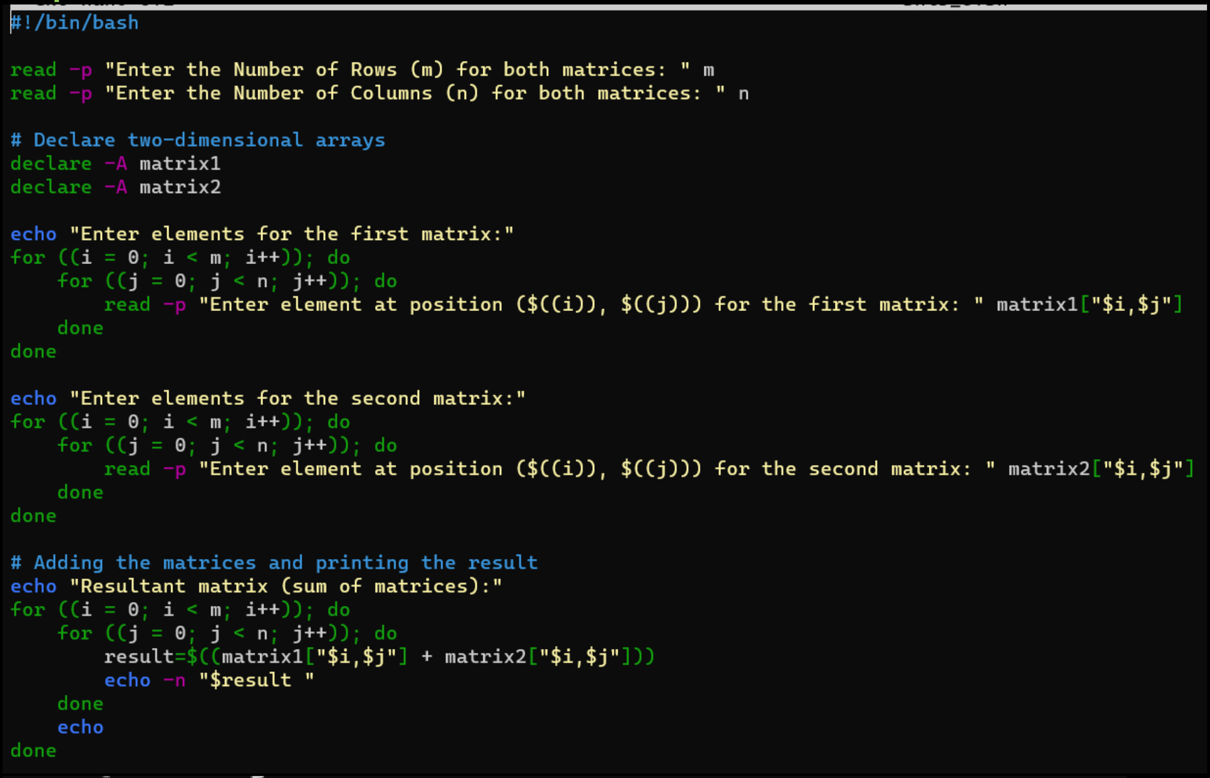


```
#!/bin/bash

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


echo -e "\n\n\n"

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
```

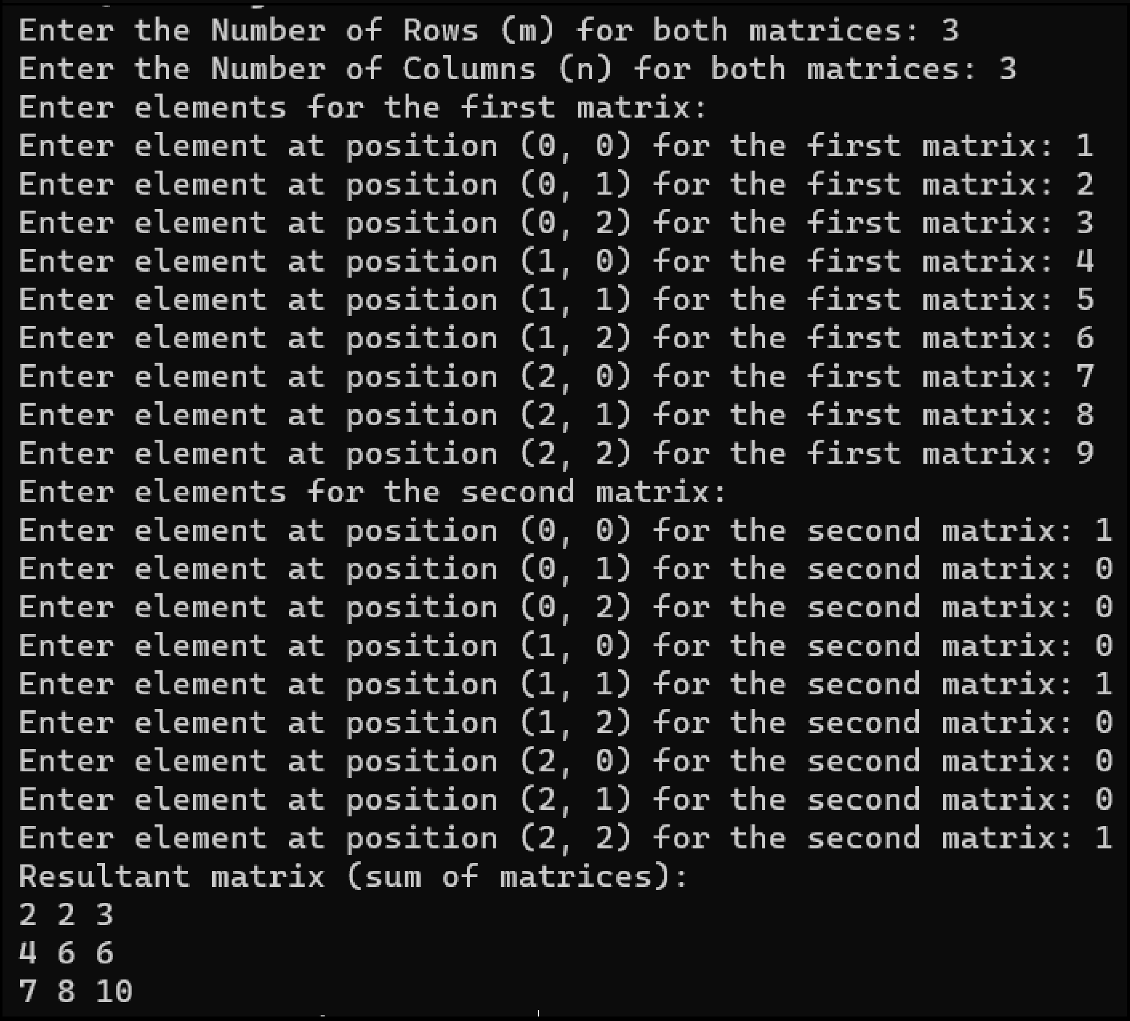


```
root@Giriirig:~/Labsheet3# ./shl3_7.sh
*****
****
***
**
*




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

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

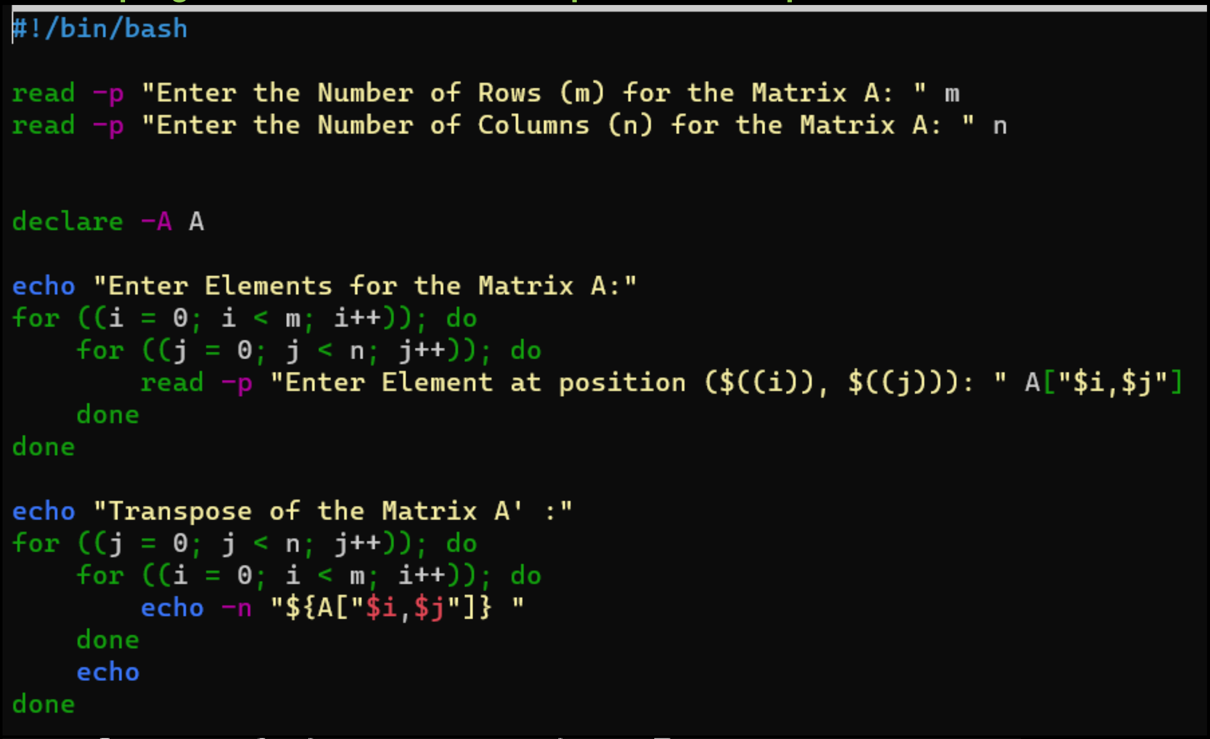


```
#!/bin/bash

read -p "Enter the Number of Rows (m) for both matrices: " m
read -p "Enter the Number of Columns (n) for both matrices: " n

# Declare two-dimensional arrays
declare -A matrix1
declare -A matrix2

echo "Enter elements for the first matrix:"
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 the second matrix:"
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

# Adding the matrices and printing the result
echo "Resultant matrix (sum of matrices):"
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
```

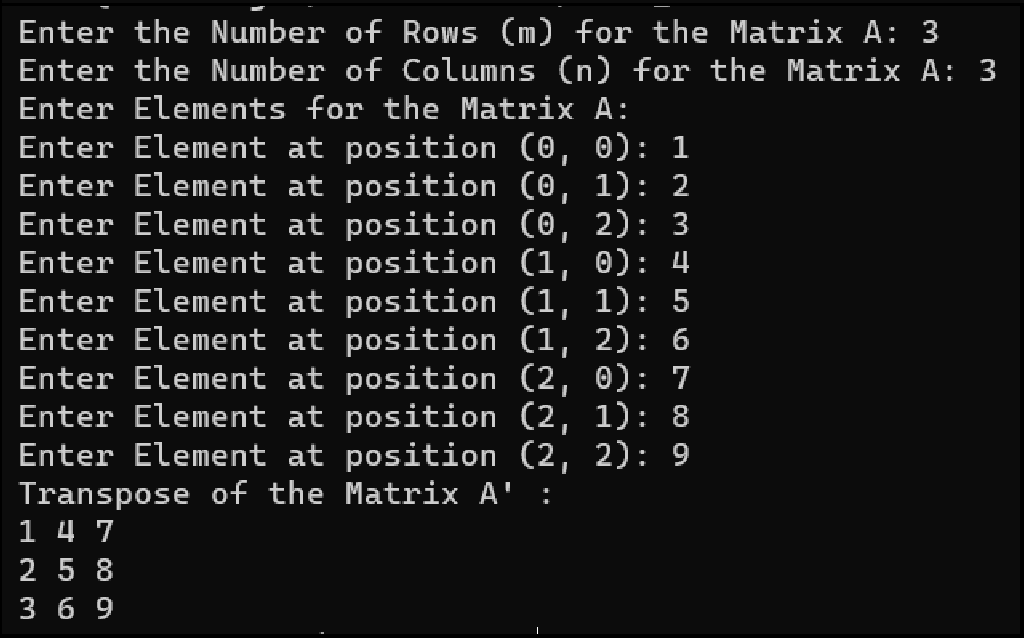


```
Enter the Number of Rows (m) for both matrices: 3
Enter the Number of Columns (n) for both matrices: 3
Enter elements for the first matrix:
Enter element at position (0, 0) for the first matrix: 1
Enter element at position (0, 1) for the first matrix: 2
Enter element at position (0, 2) for the first matrix: 3
Enter element at position (1, 0) for the first matrix: 4
Enter element at position (1, 1) for the first matrix: 5
Enter element at position (1, 2) for the first matrix: 6
Enter element at position (2, 0) for the first matrix: 7
Enter element at position (2, 1) for the first matrix: 8
Enter element at position (2, 2) for the first matrix: 9
Enter elements for the second matrix:
Enter element at position (0, 0) for the second matrix: 1
Enter element at position (0, 1) for the second matrix: 0
Enter element at position (0, 2) for the second matrix: 0
Enter element at position (1, 0) for the second matrix: 0
Enter element at position (1, 1) for the second matrix: 1
Enter element at position (1, 2) for the second matrix: 0
Enter element at position (2, 0) for the second matrix: 0
Enter element at position (2, 1) for the second matrix: 0
Enter element at position (2, 2) for the second matrix: 1
Resultant matrix (sum of matrices):
2 2 3
4 6 6
7 8 10
```

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



```
#!/bin/bash

read -p "Enter the Number of Rows (m) for the Matrix A: " m
read -p "Enter the Number of Columns (n) for the Matrix A: " n


declare -A A

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



```
Enter the Number of Rows (m) for the Matrix A: 3
Enter the Number of Columns (n) for the Matrix A: 3
Enter Elements for the Matrix A:
Enter Element at position (0, 0): 1
Enter Element at position (0, 1): 2
Enter Element at position (0, 2): 3
Enter Element at position (1, 0): 4
Enter Element at position (1, 1): 5
Enter Element at position (1, 2): 6
Enter Element at position (2, 0): 7
Enter Element at position (2, 1): 8
Enter Element at position (2, 2): 9
Transpose of the Matrix A' :
1 4 7
2 5 8
3 6 9
```