MEGHNAD SAHA INSTITUTE OF TECHNOLOGY

*Techno Complex, Madurdaha,Beside NRI Complex, Post-Uchhepota, Kolkata 700 150*

LABORATORY NOTE BOOK

MAKAUT EVEN SEMESTER 2023



[BACHELOR OF COMPUTER APPLICATION]

[UNIX AND SHELL PROGRAMMING LAB (BCAC601)]

[RUPAK SARKAR]

ROLL NO: 31001221010 REGN. NO.: 213101001210001

STREAM: BCA SEMESTER: VI (6TH)

YEAR: 3RD YearSESSION: 2021-2024



MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY



MEGHNAD SAHA INSTITUTE OF TECHNOLOGY

*Techno Complex,. Madurdaha,Beside NRI Complex, Post-Uchhepota, Kolkata 700 150*

“LIST OF ASSIGNMENT/EXPERIMENT SUBMISSION DETAILS”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL.**  **NO.** | **ASSIGNMENT / EXPERIMENT NAME** | **PLANNED DATE OF SUBMISSION** | **ACTUAL DATE OF SUBMISION** | **CHECKED BY** | **REMARKS (ANY DEVIATION REGARDING SUBMISSION DATES, CONTENT, FORMAT, ETC)** |
| 1. | Shell script to find max of three numbers. | 30/04/2024 | 30/04/2024 |  |  |
| 2. | Shell script to find SUM, AVERAGE and GRADE from marks of FOUR subjects. | 30/04/2024 | 30/04/2024 |  |  |
| 3. | Shell Script to find factorial of a number. | 30/04/2024 | 30/04/2024 |  |  |
| 4. | Sum of Digits and Total number of digits. | 30/04/2024 | 30/04/2024 |  |  |
| 5. | Shell script to find Palindrome Number. | 30/04/2024 | 30/04/2024 |  |  |
| 6. | Shell script to find Prime Number. | 30/04/2024 | 30/04/2024 |  |  |
| 7. | Shell Script to make a Menu Driven Calculator. | 30/04/2024 | 30/04/2024 |  |  |
| 8. | Shell Script to change Temperatures. | 30/04/2024 | 30/04/2024 |  |  |
| 9. | Shell Script to find Armstrong Number. | 30/04/2024 | 30/04/2024 |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| OBSERVATIONS / COMMENTS ON THE OVERALL PERFORMANCE : |

Signature in full with date Signature in full with date

**Faculty / Technical Assistant Lab Examiner**

**Q.1 Write a Shell Script to find max value from three numbers.**

#!/bin/bash

echo "Enter three integer numbers:"

read num1

read num2

read num3

max=$num1

if [ $num2 -gt $max ]; then

max=$num2

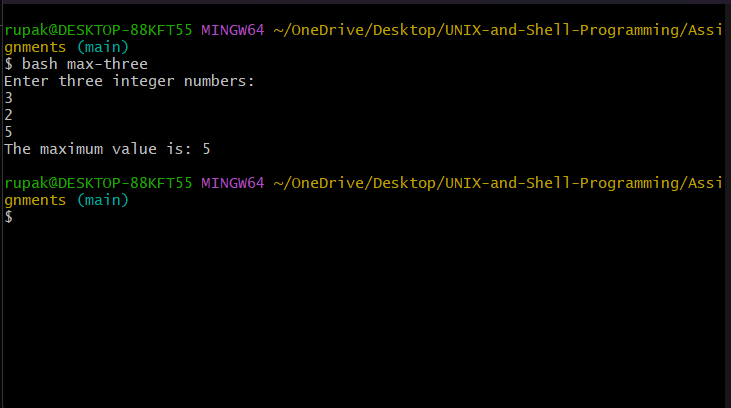
fi

if [ $num3 -gt $max ]; then

max=$num3

fi

echo "The maximum value is: $max"



**Q.2. Write a shell script to find SUM, AVERAGE and GRADE from marks of FOUR subjects.**

#!/bin/bash

calculate\_grade() {

local average=$1

local grade=""

if (( $(echo "$average >= 90" | bc -l) )); then

grade="A"

elif (( $(echo "$average >= 80" | bc -l) )); then

grade="B"

elif (( $(echo "$average >= 70" | bc -l) )); then

grade="C"

elif (( $(echo "$average >= 60" | bc -l) )); then

grade="D"

else

grade="F"

fi

echo "$grade"

}

echo "Enter marks for four subjects (separated by space):"

read mark1 mark2 mark3 mark4

sum=$((mark1 + mark2 + mark3 + mark4))

average=$(echo "scale=2; $sum / 4" | bc)

grade=$(calculate\_grade $average)

echo "Sum of marks: $sum"

echo "Average marks: $average"

echo "Grade: $grade"

**Q.3. write a shell script to find factorial of a given input integer number.**

#!/bin/bash

factorial() {

local n=$1

local result=1

for ((i=1; i<=n; i++)); do

result=$((result \* i))

done

echo "$result"

}

echo "Enter an integer to find its factorial:"

read number

if [ $number -lt 0 ]; then

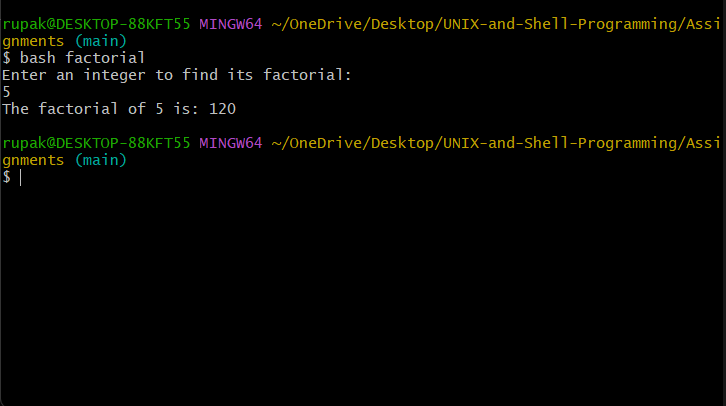
echo "Factorial is not defined for negative numbers."

exit 1

fi

result=$(factorial $number)

echo "The factorial of $number is: $result"



**Q.4. write a shell script to input a number and Find sum of digits, count the total number of digits.**

#!/bin/bash

# Function to find sum of digits and count total number of digits

sum\_and\_count\_digits() {

local num=$1

local sum=0

local count=0

while [ $num -gt 0 ]; do

digit=$((num % 10))

sum=$((sum + digit))

num=$((num / 10))

count=$((count + 1))

done

echo "Sum of digits: $sum"

echo "Total number of digits: $count"

}

# Prompt the user to enter a number

echo "Enter a number:"

# Read the input number

read number

# Call function to calculate sum of digits and count total number of digits

sum\_and\_count\_digits $number



**Q.5. write a shell script to input a number and Find reverse of this number and check whether this input number is PALINDROME or NOT.**

#!/bin/bash

# Function to reverse a number

reverse\_number() {

local num=$1

local reverse=0

while [ $num -gt 0 ]; do

digit=$((num % 10))

reverse=$((reverse \* 10 + digit))

num=$((num / 10))

done

echo "$reverse"

}

# Prompt the user to enter a number

echo "Enter a number:"

# Read the input number

read number

# Call function to reverse the number

reverse=$(reverse\_number $number)

# Check if the number is palindrome

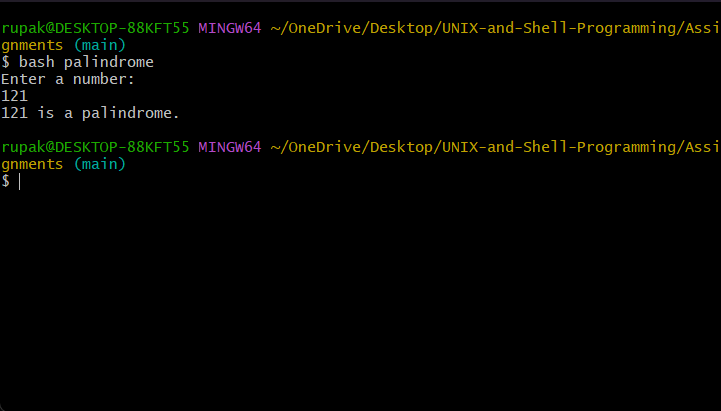
if [ $number -eq $reverse ]; then

echo "$number is a palindrome."

else

echo "$number is not a palindrome."

fi



**Q.6. Write a shell script to check whether a input number is PRIME or NOT.**

#!/bin/bash

# Function to check if a number is prime

is\_prime() {

local num=$1

local isPrime=1

if [ $num -eq 1 ]; then

isPrime=0

fi

for ((i=2; i\*i<=num; i++)); do

if [ $((num % i)) -eq 0 ]; then

isPrime=0

break

fi

done

echo $isPrime

}

# Prompt the user to enter a number

echo "Enter a number:"

# Read the input number

read number

# Call function to check if the number is prime

result=$(is\_prime $number)

if [ $result -eq 1 ]; then

echo "$number is a prime number."

else

echo "$number is not a prime number."

fi



**Q.7. Write a shell script to make a menu driven calculator with float(real) values by performing following operations:**

#!/bin/bash

# Function to perform addition

addition() {

local result=$(echo "$1 + $2" | bc)

echo "Result: $result"

}

# Function to perform subtraction

subtraction() {

local result=$(echo "$1 - $2" | bc)

echo "Result: $result"

}

multiplication() {

local result=$(echo "$1 \* $2" | bc)

echo "Result: $result"

}

# Function to perform division

division() {

local result=$(echo "scale=2; $1 / $2" | bc)

echo "Result: $result"

}

remainder() {

local result=$(echo "$1 % $2" | bc)

echo "Result: $result"

}

# Function to find square root

square\_root() {

local result=$(echo "scale=2; sqrt($1)" | bc)

echo "Result: $result"

}

power() {

local result=$(echo "$1^$2" | bc)

echo "Result: $result"

}

# Menu

echo "Menu Driven Calculator"

echo "1. Addition"

echo "2. Subtraction"

echo "3. Multiplication"

echo "4. Division"

echo "5. Remainder"

echo "6. Square Root"

echo "7. Power"

echo "0. Exit"

read choice

case $choice in

1) echo "Enter two numbers:"; read num1 num2; addition $num1 $num2 ;;

2) echo "Enter two numbers:"; read num1 num2; subtraction $num1 $num2 ;;

3) echo "Enter two numbers:"; read num1 num2; multiplication $num1 $num2 ;;

4) echo "Enter two numbers:"; read num1 num2; division $num1 $num2 ;;

5) echo "Enter two numbers:"; read num1 num2; remainder $num1 $num2 ;;

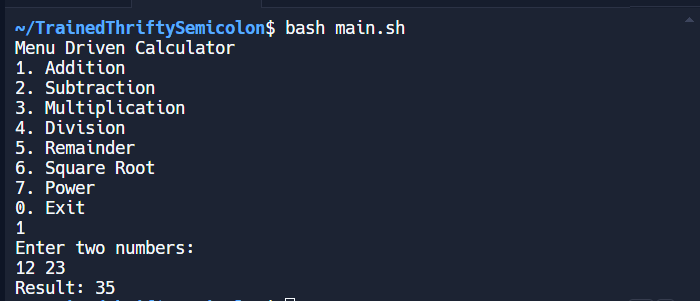
6) echo "Enter a number:"; read num; square\_root $num ;;

7) echo "Enter two numbers:"; read num1 num2; power $num1 $num2 ;;

0) echo "Exiting..."; exit ;;

\*) echo "Invalid choice"; exit 1 ;;

esac



**Q.8. Write a shell script Find temperature from 90 to 110 in Fahrenheit to Celsius and vice versa.**

#!/bin/bash

fahrenheit\_to\_celsius() {

local fahrenheit=$1

local celsius=$(echo "scale=2; ($fahrenheit - 32) \* 5 / 9" | bc)

echo "Temperature in Celsius: $celsius °C"

}

celsius\_to\_fahrenheit() {

local celsius=$1

local fahrenheit=$(echo "scale=2; $celsius \* 9 / 5 + 32" | bc)

echo "Temperature in Fahrenheit: $fahrenheit °F"

}

# Menu

echo "Temperature Conversion"

echo "1. Fahrenheit to Celsius"

echo "2. Celsius to Fahrenheit"

echo "0. Exit"

read choice

case $choice in

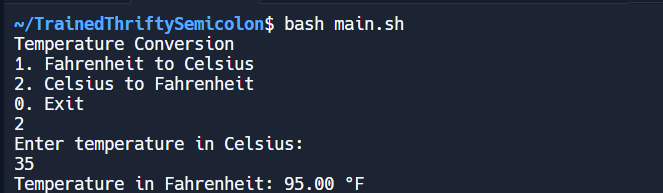
1) echo "Enter temperature in Fahrenheit:"; read temp; fahrenheit\_to\_celsius $temp ;;

2) echo "Enter temperature in Celsius:"; read temp; celsius\_to\_fahrenheit $temp ;;

0) echo "Exiting..."; exit ;;

\*) echo "Invalid choice"; exit 1 ;;

Esac



**Q.9. Write a shell script to check whether the inputted number is ARMSTRONG or NOT.**

#!/bin/bash

echo "Enter a number:"

read number

# Count the number of digits

num\_digits=${#number}

# Initialize sum to 0

sum=0

# Iterate through each digit

for (( i=0; i<$num\_digits; i++ )); do

digit=${number:$i:1}

# Calculate digit raised to the power of num\_digits and add it to sum

sum=$((sum + digit \*\* num\_digits))

done

# Check if the sum is equal to the original number

if [ $sum -eq $number ]; then

echo "$number is an Armstrong number"

else

echo "$number is not an Armstrong number"

fi

