

Experiment 4: Shell Programming

- i. Write a simple shell script that prints "Hello, World!" when executed

SOLUTION

To create a shell script that prints "**Hello, World!**", you:

1. Create a script using vi or nano.
2. Use the `echo` command to print the message.
3. Save the script with a `.sh` extension, give it execute permission, and run it.

```
echo "Hello, World!"
```

OUTPUT:

```
ubuntu@ubuntu:~$ nano helloworld.sh
ubuntu@ubuntu:~$ chmod 777 helloworld.sh
ubuntu@ubuntu:~$ nano helloworld.sh
ubuntu@ubuntu:~$ ./helloworld.sh
Hello, World!
ubuntu@ubuntu:~$ █
```

ii. Create a script that prompts the user to enter their name and then displays a personalized greeting.

SOLUTION:

1. Use the `echo` command to prompt the user to enter their name.
2. Use the `read` command to take the user's input and store it in a variable.
3. Use another `echo` to display a personalized greeting using that variable.
4. Save the script with a `.sh` extension (e.g., `greet.sh`).
5. Make it executable using `chmod +x greet.sh`.
6. Run the script using `./greet.sh` in the terminal.

```
GNU nano 7.2
echo "please enter your name"
read name
echo "hello, $person"
```

OUTPUT:

```
ubuntu@ubuntu:~$ nano name.sh
ubuntu@ubuntu:~$ chmod 777 name.sh
ubuntu@ubuntu:~$ ./name.sh
please enter your name
aditya
hello, aditya
ubuntu@ubuntu:~$
```

iii. Write a script that takes two numbers as input and performs various arithmetic operations like addition, subtraction, multiplication, and division.

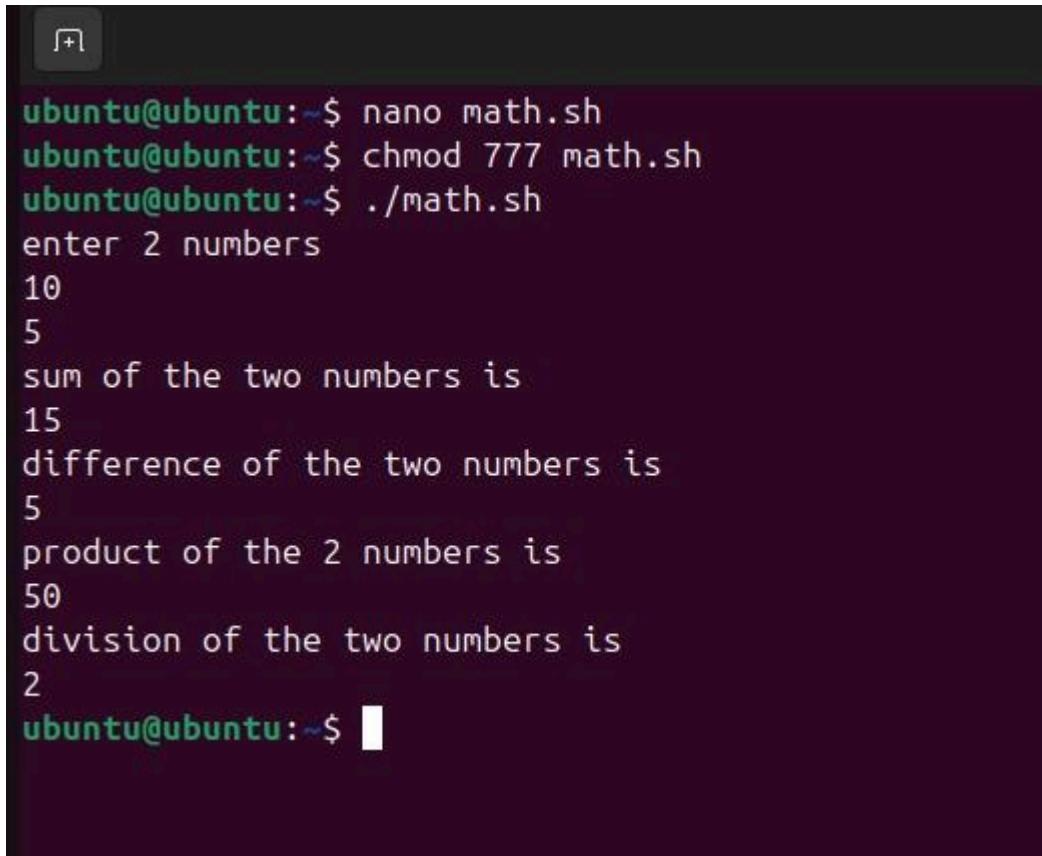
SOLUTION:

1. Use `echo` to prompt the user to enter the first number.

2. Use `read` to take the first number and store it in a variable.
3. Repeat `echo` and `read` for the second number.
4. Perform addition, subtraction, multiplication, and division using `$((. . .))` syntax for arithmetic.
5. Use `echo` to display each result.
6. Save the script with a `.sh` extension (e.g., `calc.sh`).
7. Make it executable with `chmod +x calc.sh`.
8. Run it using `./calc.sh` in the terminal.

```
GNU nano 7.2
echo "enter 2 numbers"
read num1
read num2
echo "sum of the two numbers is"
sum=$((num1 + num2))
echo "$sum"
echo "difference of the two numbers is"
difference=$((num1 - num2))
echo "$difference"
echo "product of the 2 numbers is"
product=$((num1 * num2))
echo "$product"
echo "division of the two numbers is"
division=$((num1 / num2))
echo "$division"
```

OUTPUT:

A screenshot of a terminal window on an Ubuntu system. The terminal has a dark background with light-colored text. At the top, there's a small icon of a window with a plus sign. The command line shows the user navigating to their home directory (~) and running a script named 'math.sh'. They then run the script again with the command ./math.sh. The script prompts the user to enter two numbers, which they do as 10 and 5. The script then calculates and prints the sum (15), difference (5), product (50), and division (2).

```
ubuntu@ubuntu:~$ nano math.sh
ubuntu@ubuntu:~$ chmod 777 math.sh
ubuntu@ubuntu:~$ ./math.sh
enter 2 numbers
10
5
sum of the two numbers is
15
difference of the two numbers is
5
product of the 2 numbers is
50
division of the two numbers is
2
ubuntu@ubuntu:~$
```

iv. Create a script that asks the user to enter their age and displays a message based on whether they are eligible to vote or not.

SOLUTION:

1. Use **echo** to prompt the user to enter their age.

2. Use `read` to take the input and store it in a variable.
3. Use an `if` statement to check if the age is 18 or above.
4. Use `echo` to display a message: eligible to vote if $\text{age} \geq 18$, otherwise not eligible.
5. Save the script with a `.sh` extension (e.g., `vote_check.sh`).
6. Make it executable using `chmod +x vote_check.sh`.
7. Run it using `./vote_check.sh` in the terminal.

```
GNU nano 7.2
echo "enter your age"
read age
if [ $age -lt 18 ]; then
echo "you are not eligible to vote"
else
echo "you are eligible to vote"
fi
```

OUTPUT:

```
ubuntu@ubuntu:~$ nano age.sh
ubuntu@ubuntu:~$ chmod 777 age.sh
ubuntu@ubuntu:~$ ./age.sh
enter your age
17
you are not eligible to vote
ubuntu@ubuntu:~$ ./age.sh
enter your age
19
you are eligible to vote
ubuntu@ubuntu:~$
```

Experiment 5: Shell Programming

- i. Write a script that takes a number as input and checks whether it is a prime number or not.

SOLUTION:

1. Use `echo` to prompt the user to enter a number.
2. Use `read` to take the input and store it in a variable.
3. Use a loop (like `for`) to check if the number has any divisors other than 1 and itself.

4. Use an `if` statement inside the loop to test for divisibility.
5. If a divisor is found, use `echo` to say it's not a prime number; otherwise, say it is prime.
6. Save the script with a `.sh` extension (e.g., `prime_check.sh`).
7. Make it executable using `chmod +x prime_check.sh`.
8. Run it using `./prime_check.sh` in the terminal.

```
GNU nano 1.2
echo "enter a number"
read num
flag=0
if [ $num -lt 2 ]; then
echo "number is not a prime number"
else
for ((i=2; i<=$num/2; i++)); do
if [ $(($num%i)) -eq 0 ]; then
flag=1
break
fi
done
if [ $flag -eq 1 ]; then
echo "$number is not a prime number"
else
echo "$number is a prime number"
fi
fi
```

OUTPUT:

```
ubuntu@ubuntu:~$ nano prime.sh
ubuntu@ubuntu:~$ chmod 777 prime.sh
ubuntu@ubuntu:~$ ./prime.sh
enter a number
7
is a prime number
ubuntu@ubuntu:~$ ./prime.sh
enter a number
10
is not a prime number
ubuntu@ubuntu:~$ ./prime.sh
enter a number
5
is a prime number
ubuntu@ubuntu:~$
```

ii. Write a script that calculates the sum of the digits of a given number.

SOLUTION:

1. Use `echo` to prompt the user to enter a number.
2. Use `read` to take the input and store it in a variable.
3. Initialize a variable to keep the sum of digits (start at 0).
4. Use a loop to extract each digit from the number (e.g., using modulo and division).
5. Add each extracted digit to the sum variable.
6. After the loop ends, use `echo` to display the total sum of digits.

7. Save the script with a `.sh` extension (e.g., `digit_sum.sh`).
8. Make it executable with `chmod +x digit_sum.sh`.
9. Run it using `./digit_sum.sh` in the terminal.

```
GNU nano 7.2
echo "enter a number"
read num
num1=0
sum=0
if [ $num -lt 10 ]; then
echo "invalid number"
else
while [ $num -gt 0 ]; do
num1=$((num%10))
sum=$((sum+num1))
num=$((num/10))
done
echo "$sum is the sum of the digits of the number"
fi
```

OUTPUT:

```
ubuntu@ubuntu:~$ nano sum.sh
ubuntu@ubuntu:~$ chmod 777 sum.sh
ubuntu@ubuntu:~$ ./sum.sh
enter a number
241
7 is the sum of the digits of the number
ubuntu@ubuntu:~$ ./sum.sh
enter a number
345891
30 is the sum of the digits of the number
ubuntu@ubuntu:~$ ./sum.sh
enter a number
8
invalid number
ubuntu@ubuntu:~$
```

iii. Create a script that checks whether a given number is an Armstrong number or not.

SOLUTION:

1. Use `echo` to prompt the user to enter a number.
2. Use `read` to take the input and store it in a variable.
3. Calculate the number of digits in the number.
4. Use a loop to extract each digit, raise it to the power of the number of digits, and add the result to a sum variable.
5. After the loop, compare the sum with the original number.

6. Use `echo` to display whether the number is an Armstrong number or not based on the comparison.
7. Save the script with a `.sh` extension (e.g., `armstrong.sh`).
8. Make it executable with `chmod +x armstrong.sh`.
9. Run it using `./armstrong.sh` in the terminal.

```
GNU nano 7.2
echo "enter a number"
read num
sum=0
temp=$num
while [ $temp -gt 0 ]; do
digit=$((temp % 10))
sum=$((sum + digit * digit * digit))
temp=$((temp / 10))
done
if [ $sum -eq $num ]; then
echo "$num is an armstrong number"
else
echo "$num is not an armstrong number"
fi
```

OUTPUT:

```
ubuntu@ubuntu:~$ nano armstrong.sh
ubuntu@ubuntu:~$ chmod 777 armstrong.sh
ubuntu@ubuntu:~$ ./armstrong.sh
enter a number
153
153 is an armstrong number
ubuntu@ubuntu:~$
```