**Practical Report File**

**Unix and Linux Programming**

**(CSPC-24)**



**Computer Engg. Department**

24 APRIL, 2022

**SUBMITTED TO: SUBMITTED BY:**

Dr. M.K Murmu Name: Krish Jindal

NIT Kurukshetra Roll No.: 12012104

Section: CS-B6

**Index Table for List of the experiments**

| **Sr.**  **No.** | **Name of the Experiment** | **Page No.** |
| --- | --- | --- |
| 1 | Write a C program to check whether a given integer is prime or not. | 3-4 |
| 2 | Write a C program to find the factorial of a given number. | 4 |
| 3 | Write a shell script to develop a scientific calculator (Addition, Subtraction, Multiplication, Division, sin, cos, tan, cot, log). | 5-6 |
| 4 | Write a shell script program to check whether given no is even or odd. | 7 |
| 5 | Shell script to check whether a given file is a directory or not. | 7 |
| 6 | Shell script program to count no of files in a directory. | 8 |
| 7 | Write an awk script to develop Fibonacci series. | 8 |
| 8 | Write an awk script to display the pattern of given string or number. | 9 |
| 9 | Write a program to create a zombie process. | 9-10 |
| 10 | Write a program to create a child process and allow the parent to display “parent” and the child to display “child” on screen. | 10-11 |
| 11 | Write a shell script program to include a verbose Debug option for debugging. | 11-12 |
| 12 | Write a shell script program to include xtrace debug option for debugging. | 12-13 |

**Experiment No.: 01**

**Name of the Experiment:**Write a C program to check whether a given integer is prime or not.

**Aim of the Experiment:** Check for prime integer.

**Source Code:**#include <stdio.h>

int main() {

int n, i, flag = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

for (i = 2; i <= n / 2; ++i) {

// if n is divisible by i, then n is not prime

// change flag to 1 for non-prime number

if (n % i == 0) {

flag = 1;

break;

}

}

// 0 and 1 are not prime numbers

if (n == 0 || n == 1) {

printf("%d is neither prime nor composite.\n", n);

}

else {

// flag is 0 for prime numbers

if (flag == 0)

printf("%d is a prime number.\n", n);

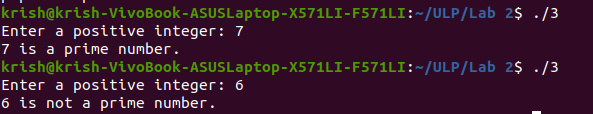
else

printf("%d is not a prime number.\n", n);

}

return 0;

}

**Sample Output:**

**Experiment No.: 02**

**Name of the Experiment:**Write a C program to find the Factorial of a given number.

**Source Code:**

#include <stdio.h>

int main() {

int n, i;

unsigned long long fact = 1;

printf("Enter an integer: ");

scanf("%d", &n);

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

fact \*= i;

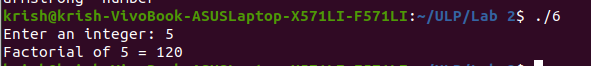
}

printf("Factorial of %d = %llu\n", n, fact);

return 0;

}

**Sample Output:**

****

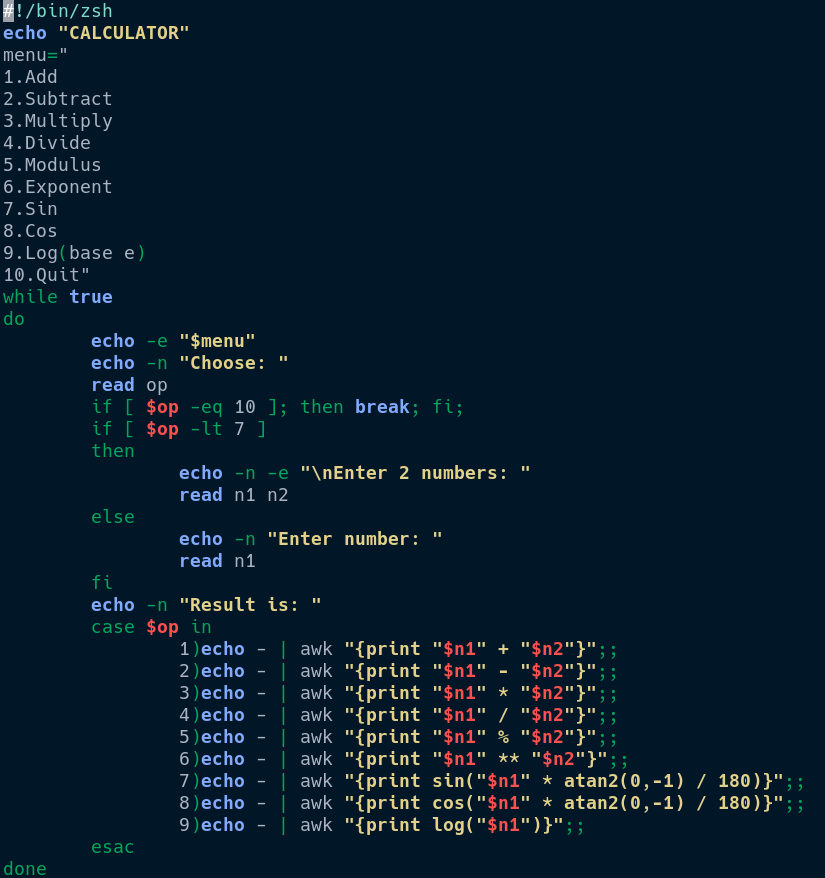
**Experiment No.: 03**

**Name:** Write a shell script to develop a scientific calculator (Addition, Subtraction, Multiplication,

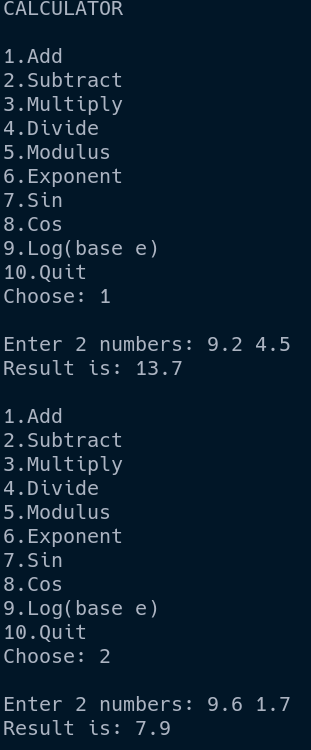
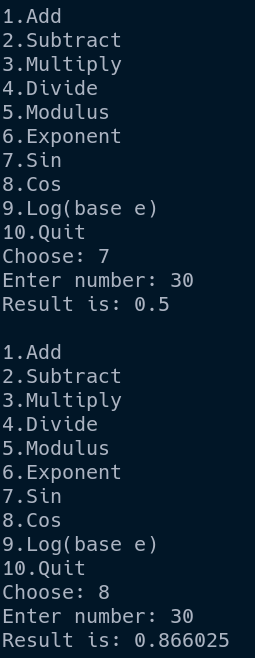
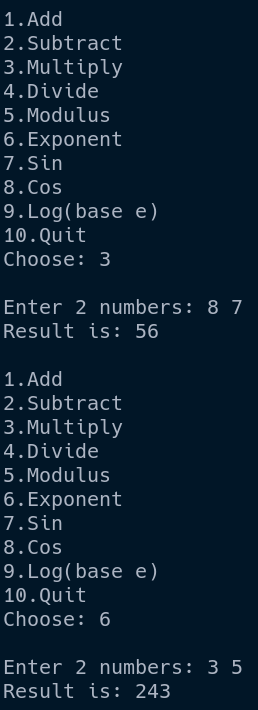
Division, sin, cos).

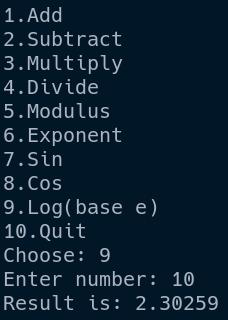
**Aim:** Make a scientific calculator.

**Source Code:**

****

**Sample Output:**

** **

****

**Experiment No.: 4**

**Write a Shell script program to check whether a number is even or odd.**

# !/bin/bash

# Take user Input

echo "Enter the number : "

read n

echo "Result:"

if [ `expr $n % 2` == 0 ]

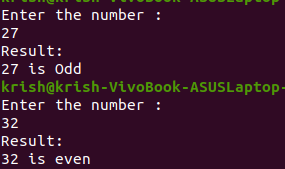
then

echo "$n is even"

else

echo "$n is Odd"

fi

****

**Experiment No.: 5**

**Shell script Program to check whether a given file is a directory or not.**

#!/bin/bash

echo "Enter details to search"

read file

if [ -d $file ];

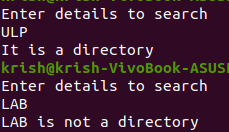
then

echo "It is a directory"

else

echo "$file is not a directory"

fi



**Experiment No.: 6**

**Shell script Program to count the number of files in a directory.**

# !/bin/bash

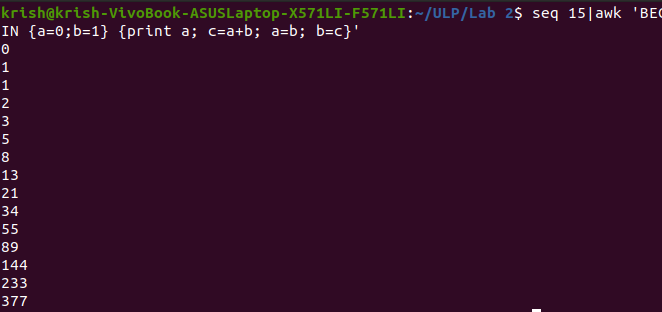
echo "Number of file present are: "

echo $(ls | wc -l)



**Experiment No.: 7**

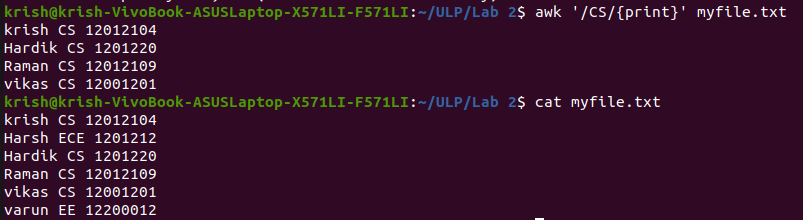
**Write an awk script to develop Fibonacci series.**



**Experiment No.: 8**

**Name: Write an awk script to display the pattern of given string or number.**

**Aim: Display the pattern of a string or a number**



**Experiment No.: 9**

**Name:** Write a program to create a zombie process.

**Aim:** Create a zombie process

**Source Code:**

#include <stdlib.h>

#include <sys/types.h>

#include <unistd.h>

int main ()

{

pid\_t child\_pid;

child\_pid = fork ();

if (child\_pid > 0) {

sleep (60);

}

else {

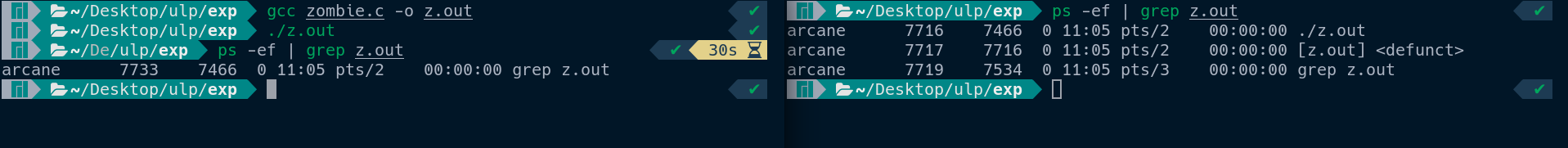
exit (0);

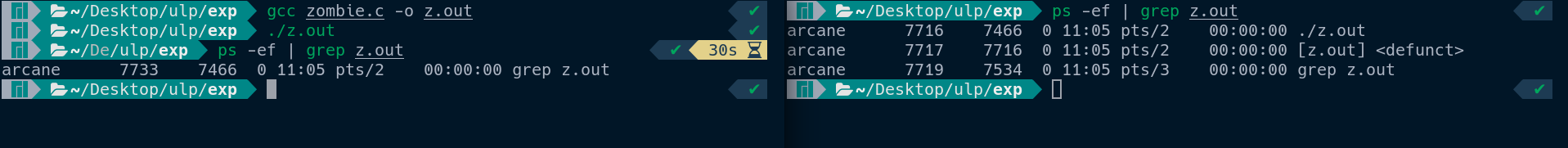
}

return 0;

}

**Sample Output:**

****

****

**Experiment No.: 10**

**Write a program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.**

**Source Code:**

#include <stdio.h>

#include <stdlib.h>

#include <sys/wait.h> /\* contains prototype for wait \*/

int main(void)

{

int pid;

int status;

printf("Hello World!\n");

pid = fork();

if (pid == -1) /\* check for error in fork \*/

{

perror("bad fork");

exit(1);

}

if (pid == 0)

printf("Child\n");

else

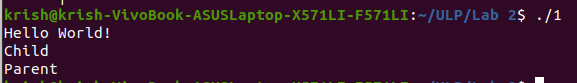
{

wait(&status); /\* parent waits for child to finish \*/

printf("Parent\n");

}

}



**Experiment No.: 11**

**Name:** Write a shell script program to include a verbose debug option for debugging.

**Aim:** Include a verbose debug option for debugging

**Source Code:**

#!/bin/bash

set -v

echo "Enter a number: "

read n1

echo "Enter another number: "

read n2

if [ "$n1" -gt "$n2" ]

then

echo "$n1>$n2"

elif ["$n1" -eq "$n2"]

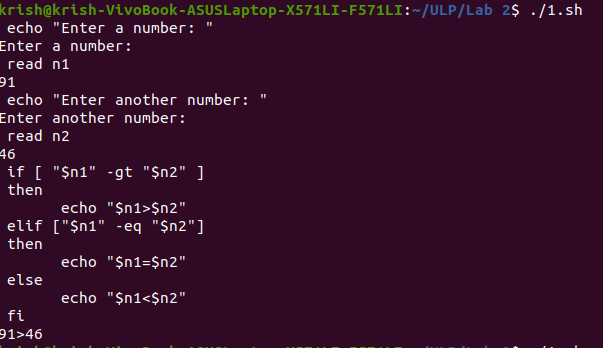
then

echo "$n1=$n2"

else

echo "$n1<$n2"

fi



**Experiment No.: 12**

**Name:** Write a shell script program to include xtrace debug option for debugging.

**Aim: I**nclude a xtrace debug option for debugging

**Source Code:**

#!/bin/bash

set -x

echo "Enter a number: "

read n1

echo "Enter another number: "

read n2

if [ "$n1" -gt "$n2" ]

then

echo "$n1>$n2"

elif ["$n1" -eq "$n2"]

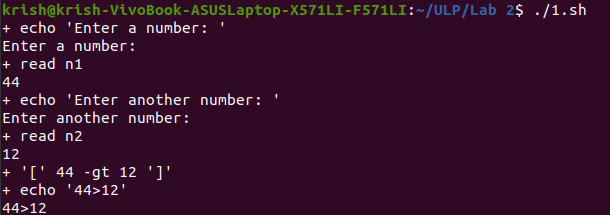
then

echo "$n1=$n2"

else

echo "$n1<$n2"

fi



-------------------END OF FILE---------------------------------------------------------------------