**Shell Programming**

Experiment No : 2 Date :-

Aim :- To implement the following shell programs:

1. Print prime numbers from 1-50 and generate its sum
2. Count numbers of files in a directory
3. Find area of rectangle, square and circle
4. Root of quadratic equation
5. Menu driven program:
6. Display present working directory
7. Display users of the system
8. Display calendar of year input through keyboard

Theory :

Shell script

A shell script is a computer program written in a scripting language that is interpreted by a command-line shell. The primary purpose of a shell script is to automate tasks and execute a series of commands in a specific sequence.

1. If else statement

if-else statement in shell programming is a control structure used for making decisions based on certain conditions. It allows you to execute different blocks of code depending on whether a given condition is true or false.

Syntax :

if [ condition is true ]

then

# Commands to execute if the condition is true

else

# Commands to execute if the condition is false

fi

Example :

echo "Please enter your age: "

read age

if [ "$age" -ge 18 ]; then

echo "You are an adult."

else

echo "You are a minor."

fi

1. while loop

while loop is used to repeatedly execute a block of code as long as a certain condition is true.

Syntax :

while [ condition is true ]

do

# Commands to execute if the condition is true

done

Example :

counter=1

while [ $counter -le 5 ]

do

echo $counter

((counter++))

Done

1. for loop

for loop is used to iterate over a sequence of values (usually a list of items) and perform a set of commands for each value in the sequence.

Syntax :

for i in {1..n}

do

# Commands to execute

done

for((c = 1; c <= n; c++))

do

# Commands to execute

done

Example :

n=5

for i in {1..$n}

do

echo "Iteration $i"

done

1. case statement

case statement is used for conditional branching based on the value of a variable. It is a more versatile alternative to using multiple if statements when you have several different cases to handle.

Syntax :

case expression in

pattern1)

# Code to execute if variable matches pattern1

;;

pattern2)

# Code to execute if variable matches pattern2

;;

pattern3)

# Code to execute if variable matches pattern3

;;

\*)

# Code to execute if variable matches none of the patterns

;;

Esac

Example :

echo "Enter a number between 1 and 3: "

read number

case $number in

1)

echo "You selected one."

;;

2)

echo "You selected two."

;;

3)

echo "You selected three."

;;

\*)

echo "Invalid input. Please enter a number between 1 and 3."

;;

esac

Code

Program 1:

for num in {1..50}

do

if [ $num -eq 1 ]; then

continue

fi

is\_prime=true

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

do

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

then

is\_prime=false

break

fi

done

if $is\_prime

then

echo $num

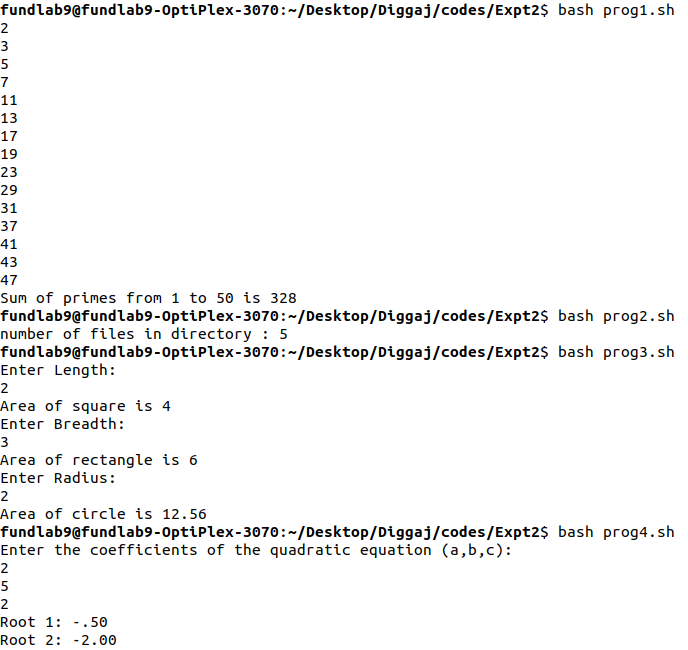
sum=$(($sum+$num))

fi

done

echo "Sum of primes from 1 to 50 is $sum"

Output 1:



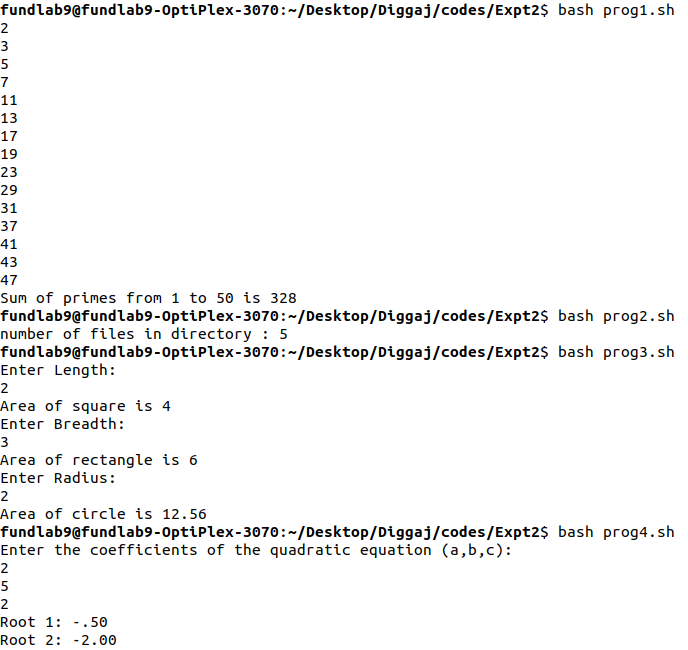
Program 2:

directory="/home/fundlab9/Desktop/Diggaj/codes/Expt2"

count=$(ls -l $directory| grep ^- | wc -l)

echo "number of files in directory : $count"

Output 2:



Program 3:

echo "Enter Length:"

read l

squ=$((l \* l))

echo "Area of square is $squ"

echo "Enter Breadth:"

read b

rec=$((l \* b))

echo "Area of rectangle is $rec"

echo "Enter Radius:"

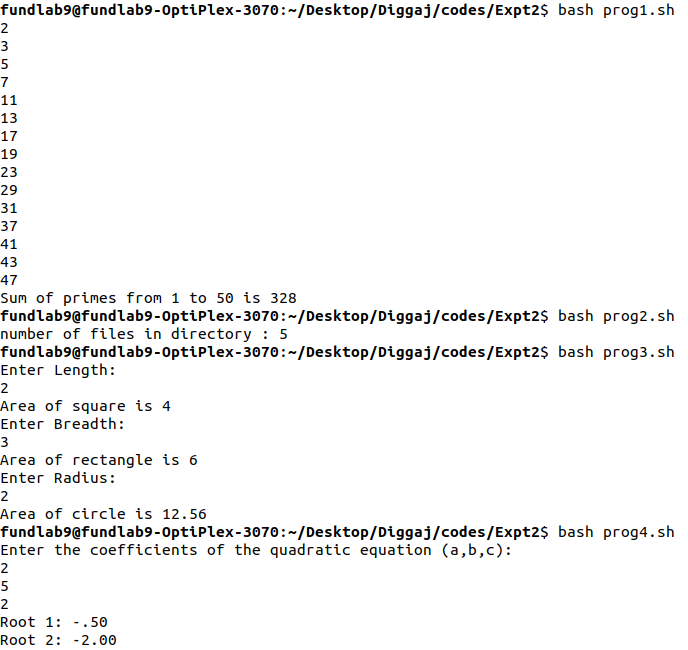
read r

pi=3.14

cir=$(echo "$pi \* $r \* $r" | bc)

echo "Area of circle is $cir"

Output 3:



Program 4:

echo "Enter the coefficients of the quadratic equation (a,b,c):"

read a

read b

read c

d=$((b\*b - 4\*a\*c))

if [ $d -gt 0 ]

then

root1=$(echo "scale=2; (-$b + sqrt($d)) / (2\*$a)" | bc)

root2=$(echo "scale=2; (-$b - sqrt($d)) / (2\*$a)" | bc)

echo "Root 1: $root1"

echo "Root 2: $root2"

else

if [ $d -eq 0 ]; then

root1=$(echo "scale=2; -$b / (2\*$a)" | bc)

echo "Root: $root1"

else

real=$(echo "scale=2; -$b / (2\*$a)" | bc)

imag=$(echo "scale=2; sqrt(-$d) / (2\*$a)" | bc)

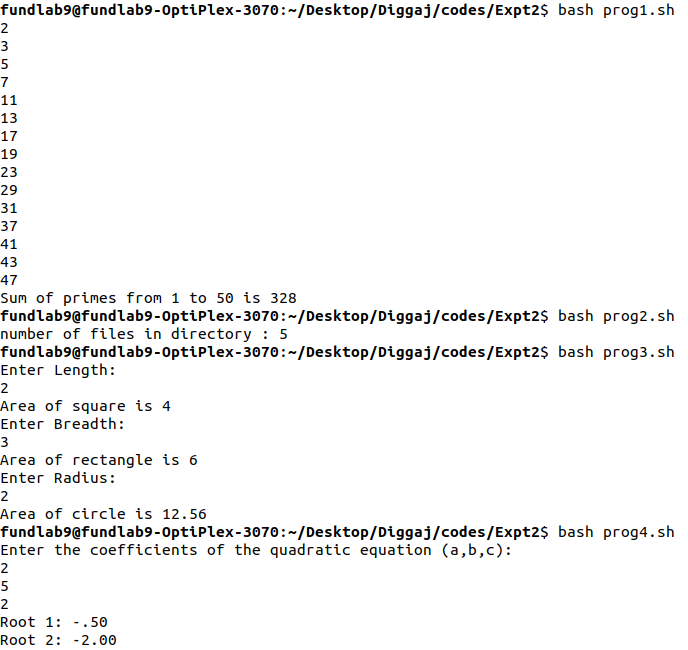
echo "Root 1: $real + $imag i"

echo "Root 2: $real - $imag i"

fi

fi

Output 4:



Program 5:

ch=1

while [ $ch -eq 1 ]

do

echo "\nMenu\n"

echo "1)Display present working directory\n2)Display users of the system\n3)Display calender of a year\n"

echo "Enter you choice "

read choice

case $choice in

1)echo $(pwd)

;;

2)echo $(who)

;;

3)echo "Enter the year"

read year

echo "\n"

echo $(cal $year)

;;

\*)echo "Invaild Choice"

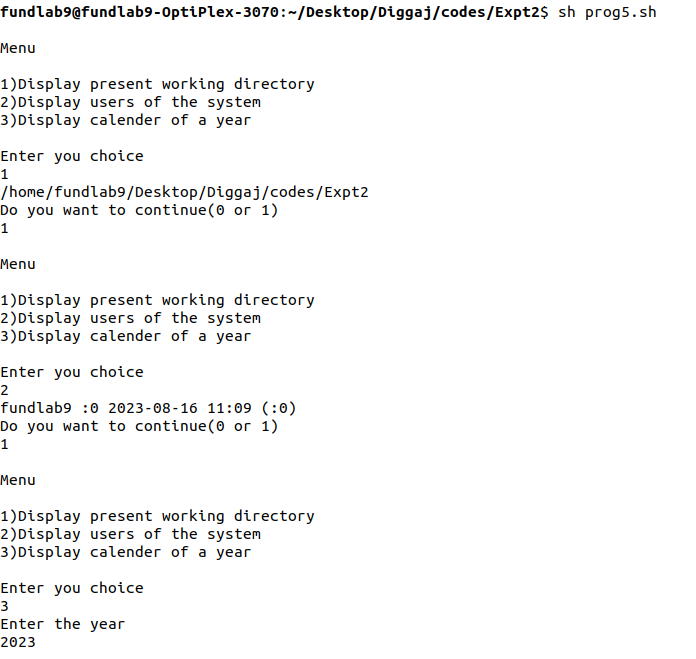
esac

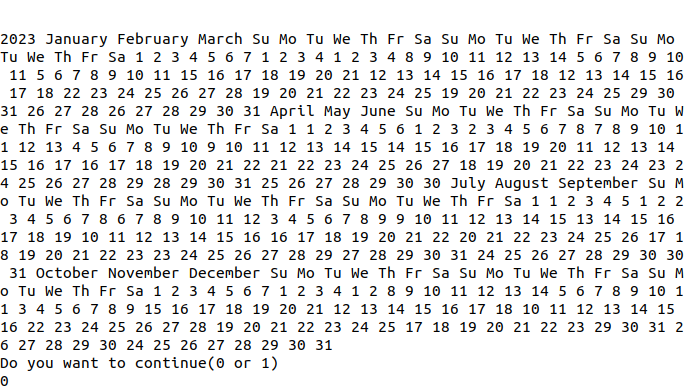
echo "Do you want to continue(0 or 1)"

read ch

done

Output 5:





**Conclusion** : Shell Programs (a), (b), (c), (d) and (e) were implemented successfully.