**Shell Programming**

1. **Program using CASE statement**

**Program :**

echo "MENU \n1. List of the files

\n2. Today's Date

\n3. Process Status

\n4. Users of system

\n5. Display Present Working Directory

\n6. Quit

\nEnter your choice : \c"

read choice

case "$choice" in

1) ls;;

2) date;;

3) ps;;

4) whoami;;

5) pwd;;

6) exit;;

\*) echo "Invalid Choice";;

Esac

**Output :**

ds14@ds14:~$ sh menu1.sh

MENU

1. List of the files

2. Today's Date

3. Process Status

4. Users of system

5. Display Present Working Directory

6. Quit

Enter your choice : 2

Monday 30 May 2022 05:19:12 PM IST

1. **Even or Odd number**

**Program :**

#Oddeven

echo "Enter a number : "

read n

rem=$(( $n % 2 ))

if [ $rem -eq 0 ]

then

echo "$n is even number"

else

echo "$n is odd number"

fi

**Output :**

Enter any number:5

Result : 5 is Odd

1. **Factorial number**

**Program:**

echo "Enter a number"

read num

fact=1

while [ $num -gt 1 ]

do

fact=$((fact \* num)) #fact = fact \* num

num=$((num - 1)) #num = num - 1

done

echo $fact

**Output:**

ds14@ds14:~$ sh factorial.sh

Enter a number

5

120

1. **Prime number or not**

**Program:**

echo "Enter a number: "

read num

i=2

f=0

while [ $i -le `expr $num / 2` ]

do

if [ `expr $num % $i` -eq 0 ]

then

f=1

fi

i=`expr $i + 1`

done

if [ $f -eq 1 ]

then

echo "The number is composite"

else

echo "The number is Prime"

fi

**Output:**

ds14@ds14:~$ sh prime.sh

Enter a number:

5

The number is Prime

1. **Fibonacci series**

**Program:**

echo "How many number of terms to be generated ?"

read n

function fib

{

x=0

y=1

i=2

echo "Fibonacci Series up to $n terms :"

echo "$x"

echo "$y"

while [ $i -lt $n ]

do

i=`expr $i + 1 `

z=`expr $x + $y `

echo "$z"

x=$y

y=$z

done

}

r=`fib $n`

echo "$r"

**Output:**

s14@ds14:~$ sh fibonacci.sh

How many number of terms to be generated ?

5

fibonacci.sh: 3: function: not found

Fibonacci Series up to 5 terms :

0

1

1

2

3

1. **Series of entered number**

**Program:**

echo "Enter Size(N)"

read N

i=1

sum=0

echo "Enter Numbers"

while [ $i -le $N ]

do

read num #get number

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

i=$((i + 1))

done

echo "Sum is : $sum"

**Output:**

ds14@ds14:~$ sh sum.sh

Enter Size(N)

5

Enter Numbers

1

2

2

1

3

Sum is : 9

1. **Power of number**

**Program:**

echo "Input number"

read no

echo "Input power"

read power

counter=0

ans=1

while [ $power -ne $counter ]

do

ans=`expr $ans \\* $no`

counter=`expr $counter + 1`

done

echo "$no power of $power is $ans"

**Output:**

Enter a interger for the base : 5

Enter a positive integer for the power : 2

25

1. **Create simple function and call function**

**Program:**

function F1()

{

echo 'I like bash programming'

}

F1

**Output:**

ds14@ds14:~$ bash fun.sh

I like bash programming

1. **Parameter function**

**Program:**

Rectangle\_Area() {

area=$(($1 \* $2))

echo "Area is : $area"

}

Rectangle\_Area 10 20

**Output:**

ds14@ds14:~$ bash para.sh

Area is : 200

1. **Function with return value**

**Program:**

function greeting() {

str="Hello, $name"

echo $str

}

echo "Enter your name"

read name

val=$(greeting)

echo "Return value of the function is $val"

**Output:**

ds14@ds14:~$ bash ret.sh

Enter your name

john

Return value of the function is Hello, john

1. **Create new directory using mkdir command**

**Program:**

echo "Enter directory name"

read newdir

`mkdir $newdir`

**Output:**

ds14@ds14:~$ gedit make.sh

ds14@ds14:~$ bash make.sh

Enter directory name

UOS

ds14@ds14:~$ ls

UOS Assign1 TYCSE