**Ex.No: 2 SHELL PROGRAMING**

**Date : 04.03.2021**

**Aim:**

To understand the shell programingusing Unix environment.

**Algorithm:**

1. Start
2. Create s file using vi command with filename.sh
3. Using ‘echo’ command , print the statement “Enter the student details”
4. Using ‘read’ command, real name,rollno,mark1,mark2,mark3 from the user
5. Using ‘expr’ command,evaluate the total marks by adding mark1,mark2,mark3 and average marks by dividing total marks by total no of marks I.e.,3
6. Using ‘echo’ command,print the student’s name,rollno,total marks,average to the user
7. Stop

**Program:**

echo "Enter name:"

read name

echo "Enter Rollno:"

read rno

echo "Enter 3 subject Marks:"

read m1

read m2

read m3

read tot

read avg

tot=`expr $m1 + $m2 + $m3`

avg=`expr $tot / 3`

echo "Academic details:

Name: $name

Roll no: $rno

Marks 1: $m1

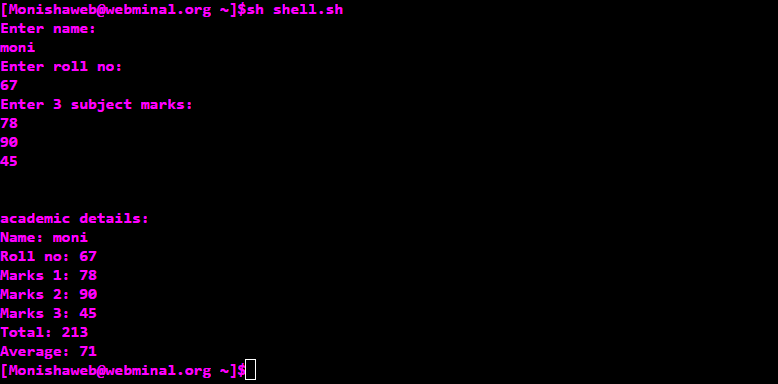
Marks 2: $m2

Marks 3: $m3

Total: $tot

Average: $avg"

**Output:**

****

1. **Implementing arithmetic operations**

**Algorithm:**

1.Create a file with extension .sh and enter into vi(virtual editor).

2.Press i to enter into insert mode

3.Prompt the user to enter a value through the keyword echo and scan them all through the keyword read

4.Perform the addition,subtraction operation

5.Display the result

6.End program

**Program:**

echo "Enter two numbers:"

read a

read b

read c

c=`expr $a + $b`

echo "Addition result: $c"

c=`expr $a - $b`

echo "Subtraction result: $c"

c=`expr $a \\* $b`

echo "Multiplication result: $c"

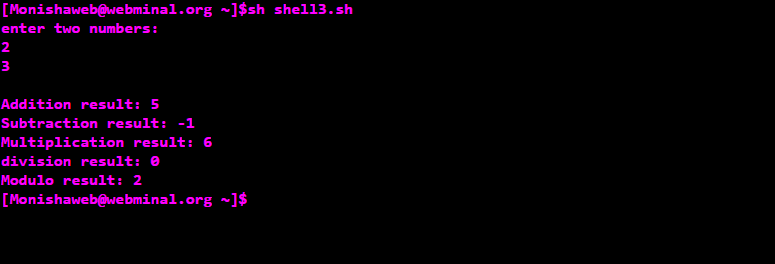
c=`expr $a / $b`

echo "Division result: $c"

c=`expr $a % $b`

echo "Modulo result: $c"

**Output:**



1. C**hecking positive,negative numbers and zeros**

**Algorithm:**

1.Create a file with extension .sh and enter into vi(virtual editor).

2.Press i to enter into insert mode

3.Prompt the user to enter a number through the keyword echo and scan it through the keyword read

4.Check whether the given number is positive or negative or equal to zero using if ,elif and else.

5.Display the result

6.End program

**Program:**

echo "Enter an integer:"

read a

if [ $a -gt 0 ]

then

echo "$a is a Positive number !! "

elif [ $a -lt 0 ]

then

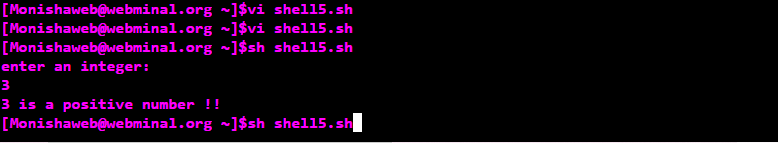
echo "$a is a Negative number !! "

else

echo "$ is ZERO !! "

fi

**Output:**

****

1. **Combinations of 1 2 3**

**Algorithm:**

1. Start

2. Create a file using vi command with filename.sh

3. Using 3 ‘for’ loops , we can print the combination of 123

4. First give ‘for var in 1 2 3’ and then give ‘do’ keyword ,like wise create 3 for loops inside one another with 3 different variables

5. Inside the third loop print the value of three variables

6. combination of 123 gets printed

7. Using ‘done’ keyword, end the each for loops one after another.

8. Stop

**Program:**

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Combinations of 1 2 3\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

a=1

b=1

c=1

for a in 1 2 3

do

for b in 1 2 3

do

for c in 1 2 3

do

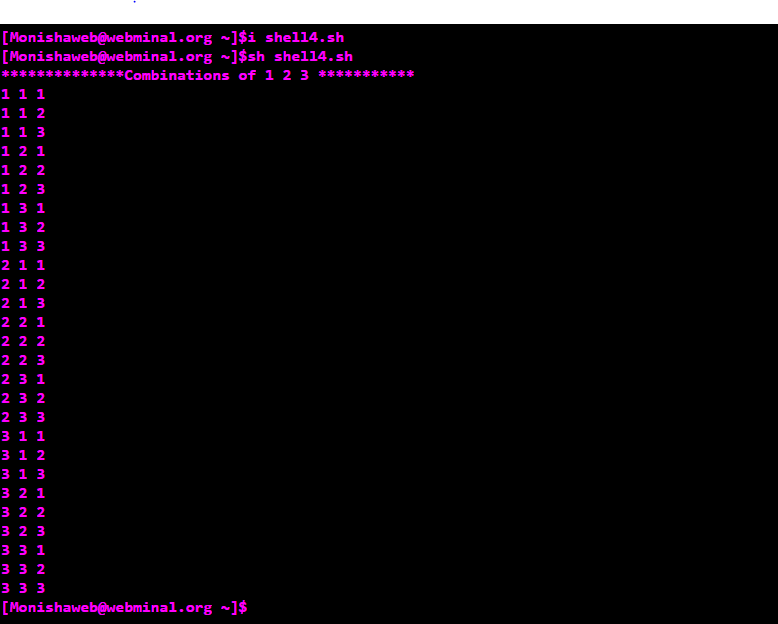
echo "$a $b $c"

done

done

done

**Output:**

****

1. **To find area of triangle,rectangle,circle,square**

**Algorithm:**

1. Start

2. Create a file using vi command with filename.sh

3. Using ‘echo’ command, print the statement “Enter your choice between 1 and 4:”

4. Using ‘read’ command, read the value of a from the user

5. Inside case, give 4 cases for triangle, circle, square, rectangle respectively. User can choose any case to execute it.

6. In first case, read the values of base and height from user and calculate and print the area of triangle.

7. In second case, read the values of radius from user and calculate and print the area of circle.

8. In third case, read the values of side from user and calculate and print the area of square.

9. In fourth case, read the values of length and breadth from user and calculate and printthe area of rectangle.

10. In default case, print “Your choice is out of limit!”

11. Terminate case with keyword ‘esac’

12. Stop

**Program:**

echo "Rectangle:

Enter length and breadth:"

read l

read b

area=`expr $l \\* $b`

echo "Area of rectangle: $area"

;;

2) echo "Square:

Enter the side:"

read aa

area=`expr $aa \\* $aa`

echo "Area of square: $area"

;;

3) echo "Circle:

Enter radius:"

read rr

area=`expr $rr \\* $rr \\* 22 / 7`

echo "Area of circle: $area"

;;

4) echo "Triangle:

Enter base and height:"

read b

read h

area=`expr $b \\* $h / 2`

echo "Area of triangle: $area"

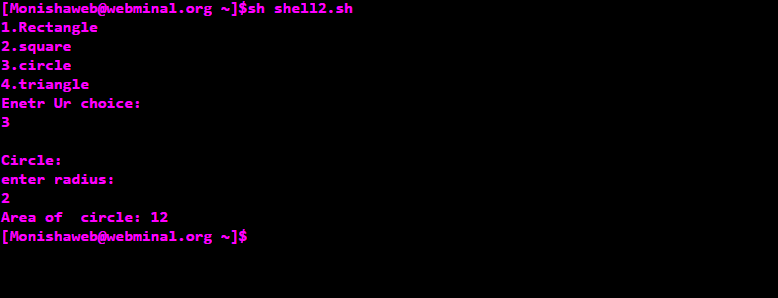
;;

\*) echo "Enter valid choice :("

;;

Esac

**Output:**

****

1. **Concatenate strings**

**Algorithm:**

1.Start

2.Create a new file using vi command filename.sh

3.Get the two strings s1,s2 from the user using ‘read’ command.

4.Concatenate two strings by simply joining them and store in in the variable s3.

5.Print the concatenated string using ‘echo’ command.

6.Print the length of the concatenated string using the statement ${#s3}.

7.Stop.

**Program:**

echo "Enter the 2 strings:"

read s1

read s2

read s3

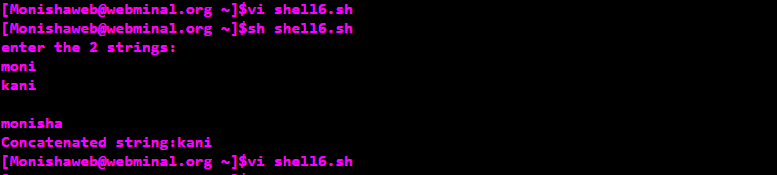
read l

s3=$s1$s2

echo "Concatenated string: $s3 "

echo "Length of the new string: ${#s3}"

**Output:**

****

1. **Printing digits at odd places in the number**

**Algorithm:**

1.Start

2.Create a new file using vi command filename.sh

3.Get the input number num from the user using ‘read’ command.

4.Declare on more variable a with value as 1.

5.Using ‘until’ loop with condition as ‘until value of a is greater than value of num’ do the step 6 and 7.

6.Print the digit in the ath coloumn digit using ‘echo’ command.

7.Increment the value of a by 2.

8.Repeat step 6 and 7 until the condition fails.

9.Stop.

**Program:**

echo "Enter the number:"

read num

read a

a=1

until [ $a -gt ${#num} ]

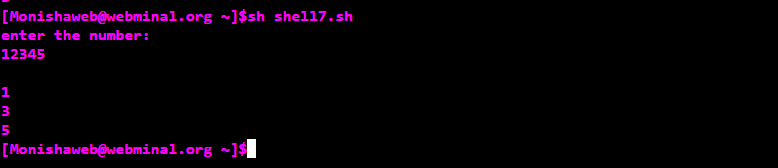
do

echo $num | cut -c $a

a=`expr $a + 2`

done

**Output:**

****

1. **Reversing a number**

**Algorithm:**

1.Start

2.Create a new file using vi command filename.sh

3. Get the input number n from the user using ‘read’ command.

4.Declare two more variables rem and rev and assign values as 0.

5.Create a temporary variable temp and store the value of n in it.

6.Using ‘while’ loop with condition ‘while value of temp is greater than 0’ do the following steps.

7.Using ‘expr’ command, evaluate the modulo of temp and 10 and store it in rem.

8.Using ‘expr’ command, evaluate the multiplication of rev and 10 and add rem with it and store it back in rem.

9.Using ‘expr’ command, evaluate the division of temp and 10 and store it in temp.

10.Repeat the steps 7,8 and 9 until the condition is satisfied.

11.Stop.

**Program:**

echo "Enter a number:"

read n

read temp

read rem

read rev

rem=0

rev=0

temp=$n

while [ $temp -gt 0 ]

do

rem=`expr $temp % 10`

rev=`expr $rev \\* 10 + $rem`

temp=`expr $temp / 10`

done

echo "Reverse of $n is : $rev"

**Output:**

****

1. **Search an element in an array**

**Algorithm:**

1.Start

2.Create a new file using vi command filename.sh

3.Declare an array a and get the size of array n from the user using ‘read’ command.

4.Declare a flag avraiable with value as 0.

5.Get the array elements from the user using ‘for’ loop.

6.Display the array using the simple statement ${a[@]}.

7.Get the key element to be searched from the user using ‘read’ command.

8.In another ‘for’ loop(which runs upto the length of the array), search for the key in the array using ‘if’ statement.

9.In ‘if’ statement, specify the equality condition as ‘if the element at ith position matches the key’, set value of flag as 1.

10. Repeat the step 9 until the loop terminates.

11.Then using ‘if’ statement, check whether the value of flag matches 1 .

12.If it is true, print “Searched element is found” else print “No such element is found”.

13.Stop.

**Program:**

declare -a a

echo "Enter the size of array:"

read n

flag=0

echo "Enter the array elements:"

for(( i=0 ; i<$n ; i++ ))

do

read ele

a[$i]=$ele

done

echo "The array: ${a[@]}"

echo "Enter the element to be searched:"

read key

for(( i=0 ; i<$n ; i++ ))

do

if [ ${a[i]} -eq $key ]

then

flag=1

fi

done

if [ $flag -eq 1 ]

then

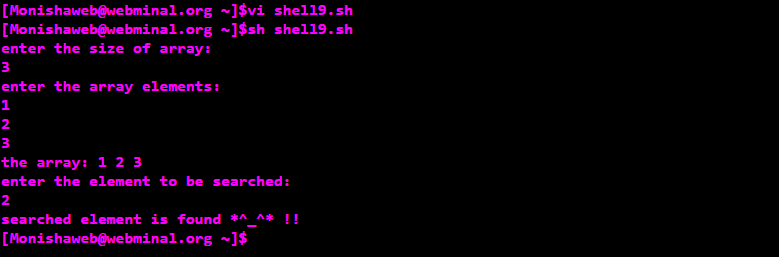
echo "Searched element is found \*^\_^\* !!"

else

echo "Ohooh!! No such element is found -\_-"

fi

**Output:**

****

1. **Deleting zero sized files**

**Algorithm:**

1.Start

2.Create a new file using vi command filename.sh

3.Use ‘for’ loop which iterate over all the files in the root directory, and do the following steps.

4.Using ‘if’ statement, check whether the file has size using –s command.

5.If the file has size greater than zero, print that the “File has size > 0”

6.Else print “File is removed as its size is 0” and remove(delete) that file using rm command.

7.Repeat the steps 4 and 5 until the loop terminates.

8.Stop.

**Program:**

echo "Deleting all ZERO SIZED files"

for a in \*

do

if [ -s $a ]

then

echo "$a has a file size > 0"

else

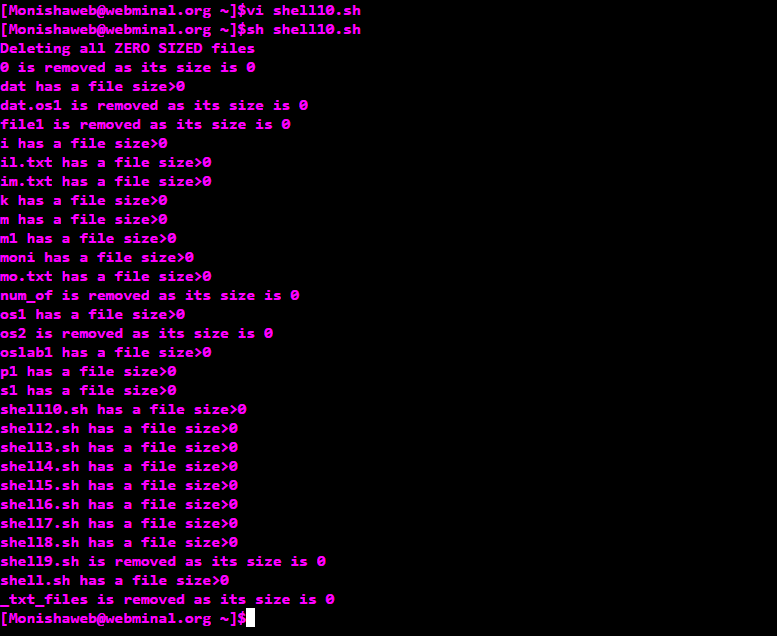
echo "$a is removed as its size is 0"

rm $a

fi

done

**Output:**

****

|  |  |
| --- | --- |
| **Observation(20)** |  |
| **Record(5)** |  |
| **Total(25)** |  |
| **Initial** |  |

**Result:**

Thus the basic Unix commands were executed and outputs were noted.