**EX NO:2 SHELL PROGRAMMING**

**DATE:04/03/2021**

**Aim:**

The aim is to execute the shell program in Linux.

**1.Getting and display the academic details. Get input from the user such as Name, Rollno, Mark1.Mark2, Mark3, Average.**

**Aim:**The aim is to get and display the academic details and to get input from the user such as Name,Rollno,Mark1.Mark2,Mark3,Average**.**

**Algorithm:**

1.Start.

2.Get the name,rollno,mark1,mark2,mark3 from the user as input.

3.Calculate the total marks using expr command and addition operation.

4.Calculate the average mark using expr command by dividing totalmarks by 3.

5.Display the name,rollno,mark1,mark2,mark3,total,average to the user by using echo command.

6.Stop.

**Program:**

echo “Enter the name”

read n

echo “Enter the rollno”

read r

echo “Enter the mark1”

read m1

echo “Enter the mark2”

read m2

echo “Enter the mark3:”

read m3

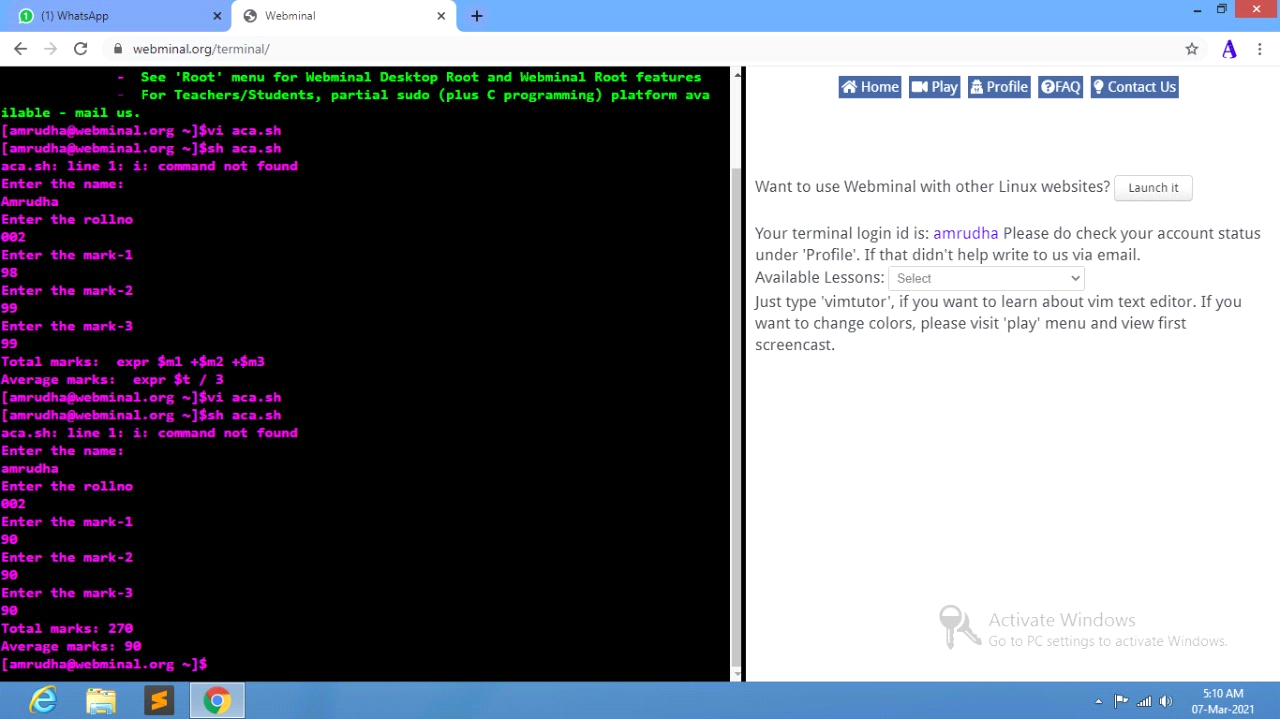
t=` expr $m1 + $m2 + m3 `

echo “Total marks :$t”

a=` expr $tot / 3 `

echo “Average marks: $a”

**output:**



**2.Display arithmetic operations in shell program in Linux.**

**Aim:**

The aim is to display arithmetic operations in shell program in linux.

**Algorithm:**

1. Start.

2. Get the value of a and b from the user.

3. Compute the sum of a and b using expr command and “+” operation.

4. Compute the difference of a and b using expr command and “-” operation.

5. Compute the product of a and b using expr command and “\\*” operation.

6. Compute the quotient of a and b using expr command and “/” operation.

7. Compute the remainder of a and b using expr command and “%” operation.

8. Stop

**Program:**

echo "Enter the value of a:"

read a

echo "Enter the value of b:"

read b

x=` expr $a + $b `

echo "sum of $a and $b is: $x"

y=` expr $a - $b `

echo "difference of $a and $b is: $y"

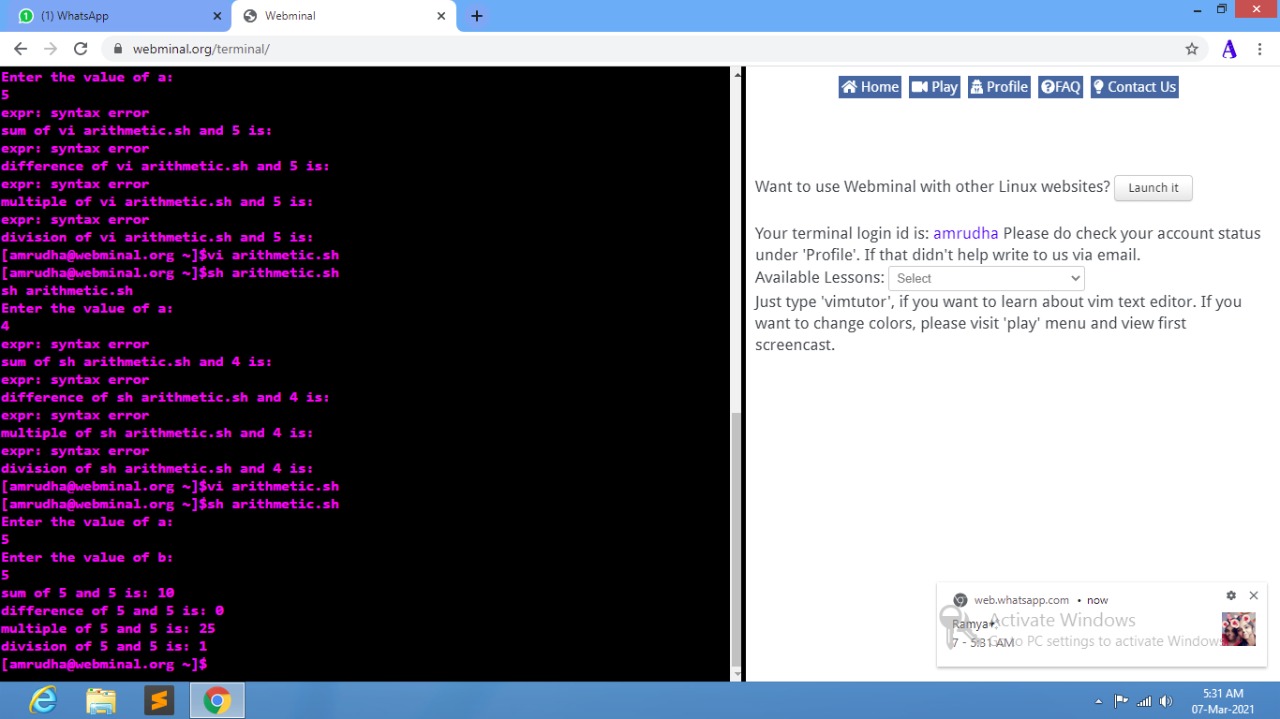
z=` expr $a \\* $b `

echo "multiple of $a and $b is: $z"

w=` expr $a / $b `

echo "division of $a and $b is: $w"

**output:**



**3.Write a shell program to check for positive and negative number**

**Aim:**

The aim is to write a shell program to check for positive and negative number.

**Algorithm:**

1. Start

2. Get a number as an input from the user.

3. If the number is greater than zero ,print the given number is positive.

4. If the number is lesser than zero ,print the given number is negative

5. If the number is equal than zero, print the given number is neither positive nor zero.

6. Stop.

**Program:**

echo "Enter the number:"

read a

if [ $a -gt 0 ]

then

echo "$a->Is the positive number"

elif [ $a -lt 0 ]

then

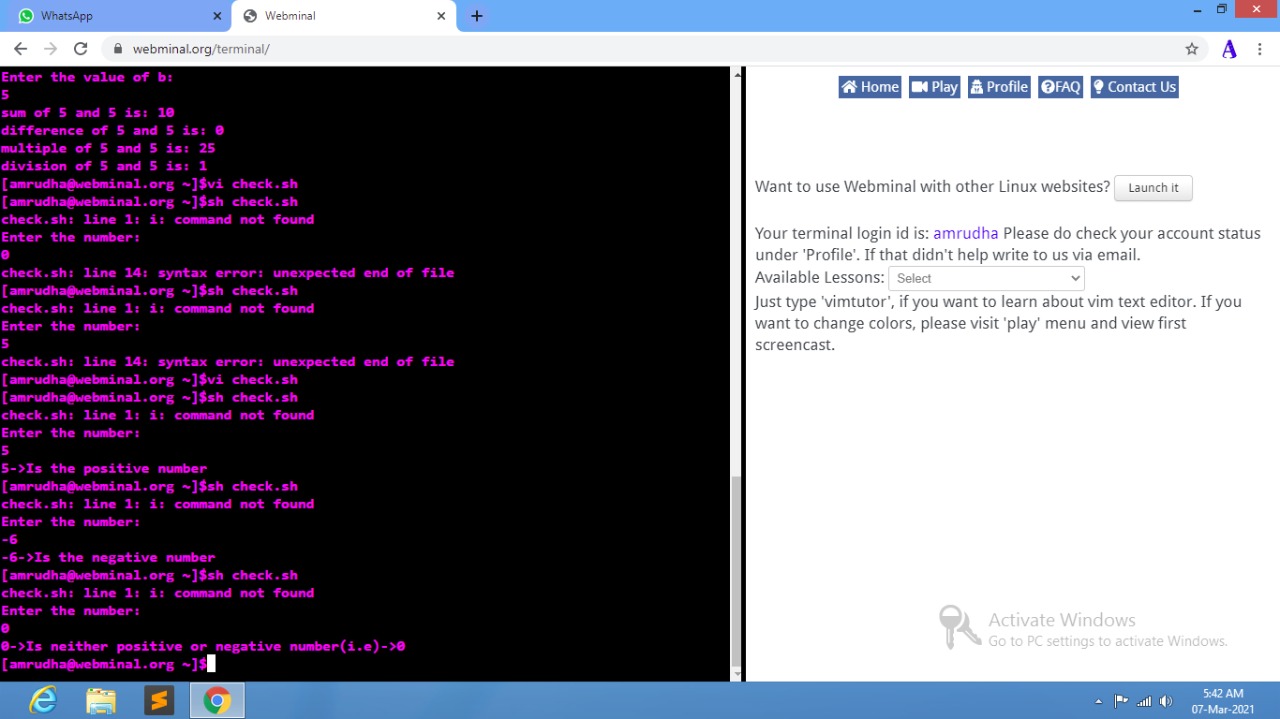
echo "$a->Is the negative number"

else

echo "$a->Is neither positive or negative number"

fi

**output:**



**4.Write a program to find the area of circle,square,,triangle,rectangle using switch case.**

**Aim:**

The aim is to write a program to find the area of circle,square,,triangle,rectangle using switch case.

**Algorithm:**

1. Start.

2. Get a choice as an input from the user.

3.If the user enters 1,get the length and breadth of the rectangle,and as the result area of the rectangle will be printed.

4.If the user enters 2,get the radius of the circle,and as the result area of the circle will be printed.

5.If the user enters 3,get the side of the square,and as the result area of the square will be printed.

6.If the user enters 4,get the height and breadth of the triangle,and as the result area of the triangle will be printed.

7.If the user enters any other choices other than this,output will be printed as “The given choice is invalid”.

8.Stop.

**Program:**

read a

case $a in

1)

echo "Enter the length of the rectangle:"

read l

echo "Enter the breadth of the rectangle:"

read b

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

echo "The area of the rectangle with length $l and breadth $b is: $a";;

2)

echo "Enter the radius of the circle:"

read r

a=` expr 22 / 7 \\* $r \\* $r `

echo " the area of the circle with radius $r is: $a";;

3)

echo "Enter the side of the square"

read s

a=` expr $s \\* $s `

echo "the area of the square with side $s is: $a";;

4)

echo "Enter the breadth of the triangle:"

read b

echo "Enter the height of the triangle:"

read h

a=`expr 1 / 2 \\* $b \\* $h`

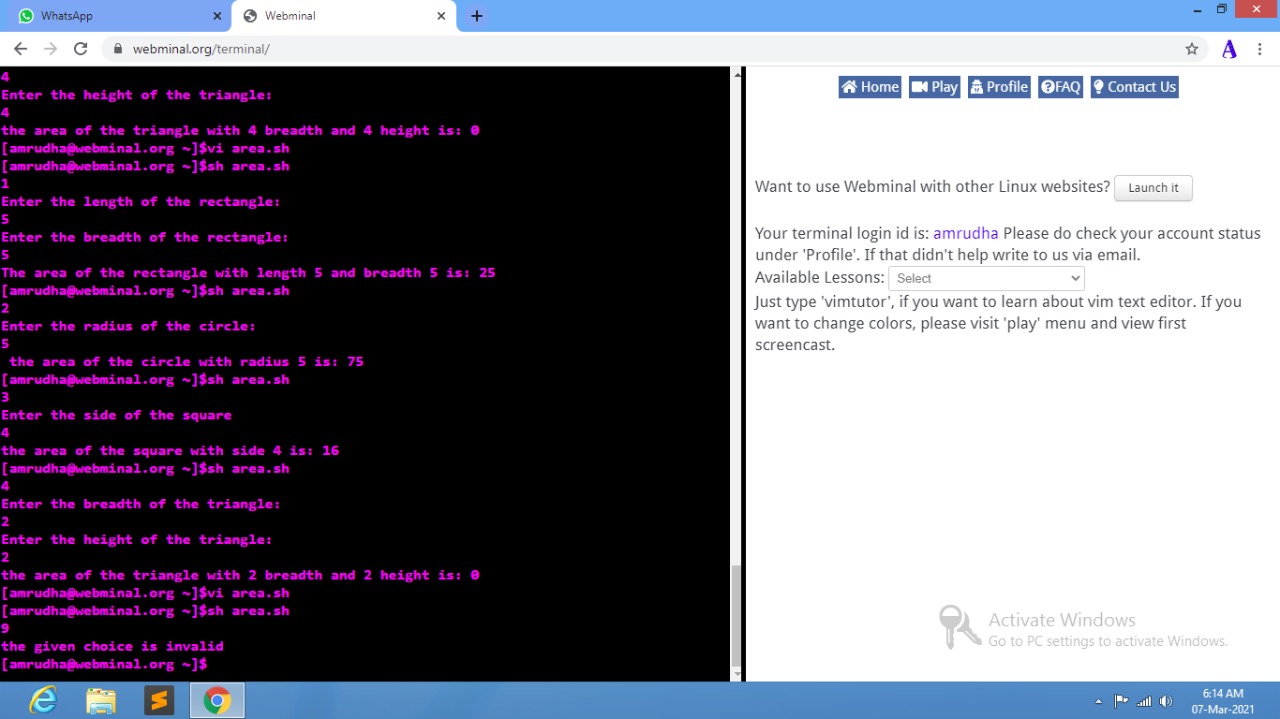
echo "the area of the triangle with $b breadth and $h height is: $a";;

\*)

echo "the given choice is invalid";;

esac

**Output:**



**5.Write a shell program to display the combination of 123:**

**Aim:**

The aim is to write a shell program to display the combination of 123

**Algorithm:**

1.Start

2.Use for loop for a in 123.

3.Use for loop for b in 123.

4. Use for loop for c in 123.

6.Finally print a ,b and c and close the for loop.

7.Stop.

**Program:**

for a in 1 2 3

do

for b in 1 2 3

do

for c in 1 2 3

do

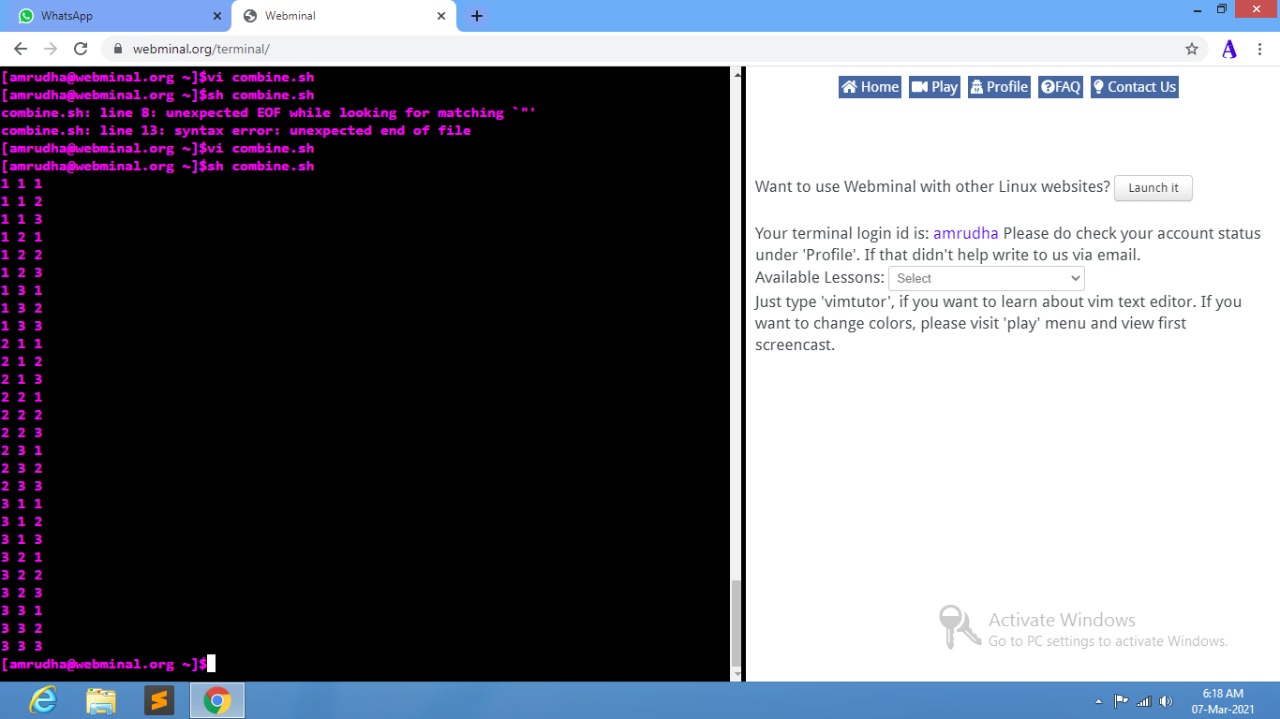
echo "$a $b $c"

done

done

done

**Output:**



**6.Write a shell program to concatenate 2 strings and find the length of the resultant string.**

**Algorithm:**

1.Start.

2.Get the first string as the input from the user.

3.Get the second string as the input from the user.

4.Concatenate two strings by joining the two strings.

5.Find the length of the string by using length command.

6.Print the length of the concatenated string.

7.Stop.

**Program:**

echo "enter the first string"

read str1

echo "enter the second string"

read str2

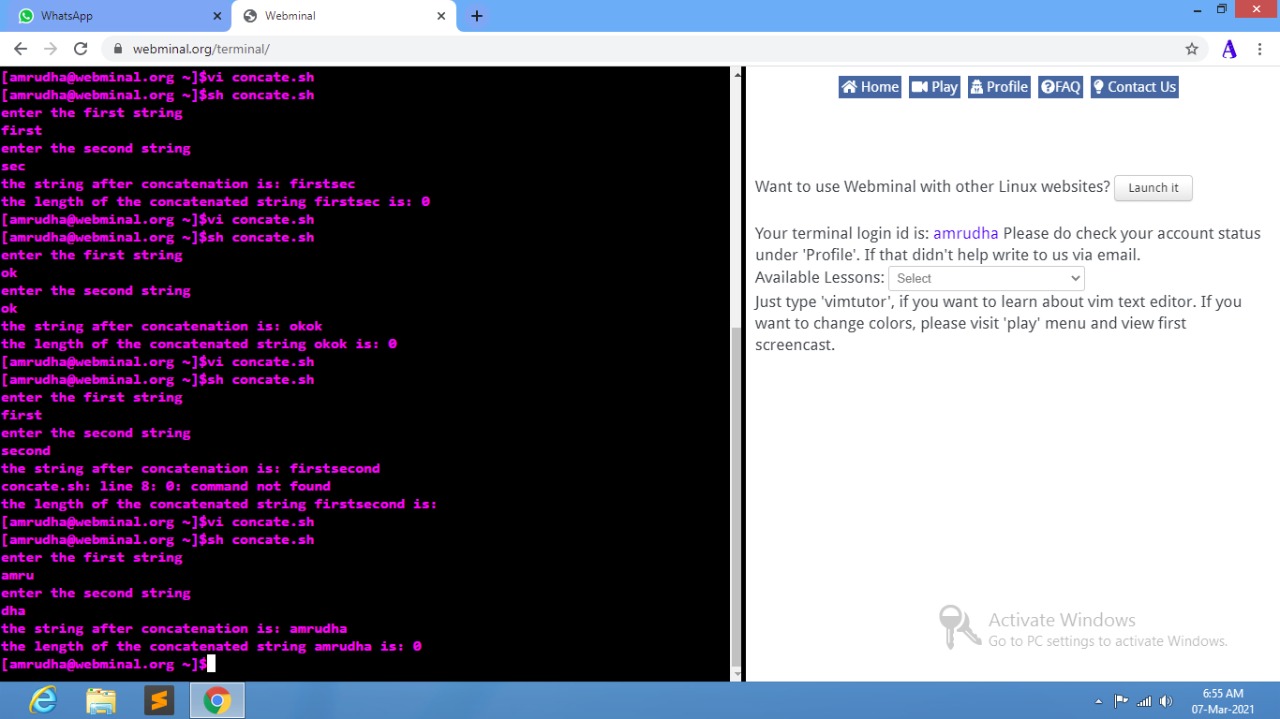
x="$str1$str2"

echo "the string after concatenation is: $x"

y=` expr length "$y" `

echo "the length of the concatenated string $x is: $y"

**Output:**



**7.Write a shell program to display the digit which are in odd position.**

**Algorithm:**

1.Start

2.Get the no of digits in a number from the user.

3.Get the required number from thre user.

4.Check for the condition that the number got from the user is less than or equal to the number of digits in the number go inside the while loop

5.The condition required is num | cut –c $n

6.Print the result as per given in the problem statement.

7.Stop.

**Program:**

echo "enter the no of digits in a number:"

read a

echo "enter the $a digit number:"

read no

n=1

while [ $n -le $a ]

do

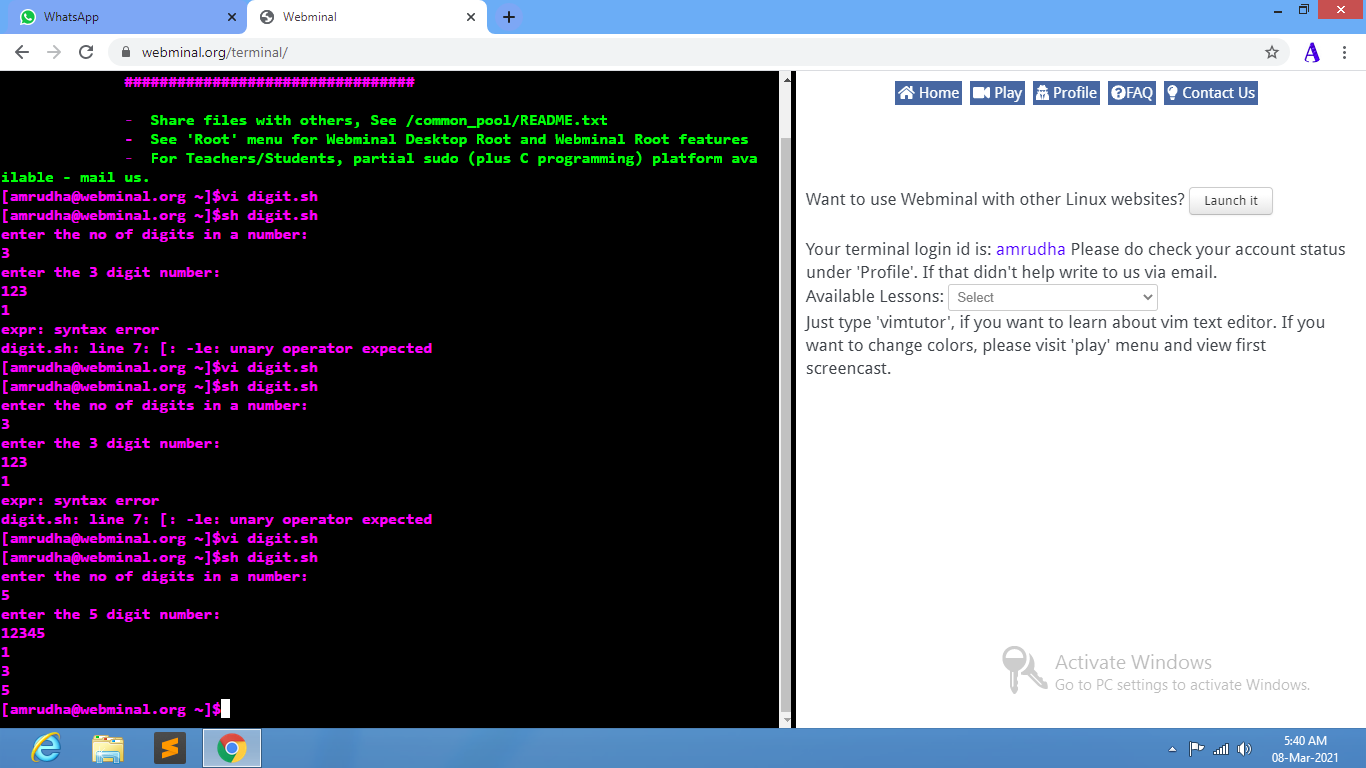
b=` echo $no |cut -c $n `

echo "$b"

n=` expr $n + 2 `

done

**output:**



**8.Write the shell program to search an element in an array:**

**Algorithm:**

1.Start.

2.Get the size of the array from the user.

3.Get the array elements from the user using while loop.

4.Get the element to be searched from the user.

5.Initialize i as 0.

6.If i is greater than 0,check for the key.

7.If the key is present in the array, print that the key is present.

8.If the key is not present in the array,print that the key is not present in the array.

8.Stop.

**Program:**

echo "enter the size of the array:"

read a

k=1

i=0

while [ $i -ne $a ]

do

echo "enter the array elements:"

read val

arr[$i]=$val

i=` expr $i + 1 `

done

echo "enter the elements to be searched"

read key

i=0

while [ $i -lt $a ]

do

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

then

k=0

else

l=0

fi

i=` expr $i + 1 `

done

if [ $k -eq 1 ]

then

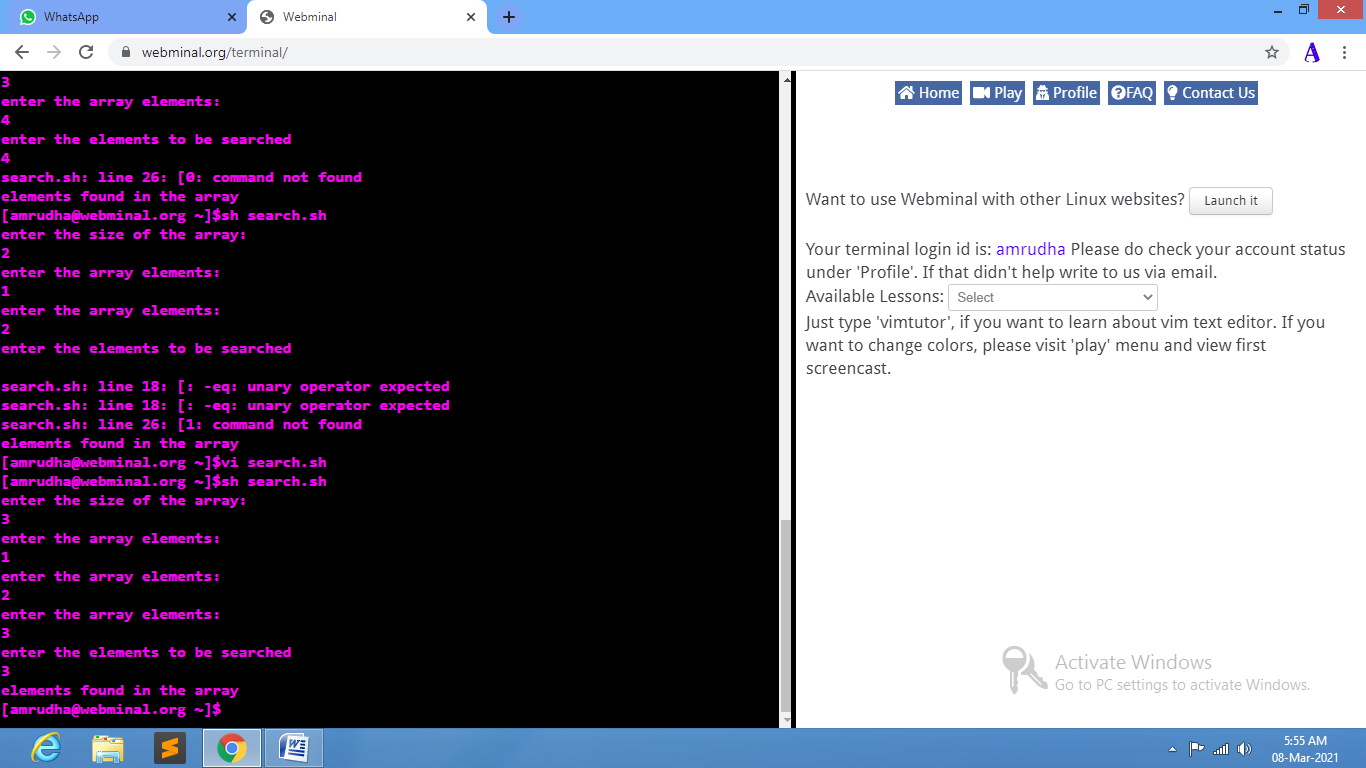
echo "elements not found in the array"

else

echo "elements found in the array"

fi

**Output:**



**9.Write the shell program to delete the zero sized file using if and for loop**

**Algorithm:**

1.Start.

2.Check for all the files using for loop and with the help of \* operator.

3.To check the empty file,use the command –s along with all the files present.

4.If the file present is empty that it is the empty file and remove the file using rm –rf command.

5.If the file present is not empty that it is not empty file.

6.Stop.

**Program:**

for i in \*

do

if [ -s $i ]; then

echo "$i is a file which is not empty"

else

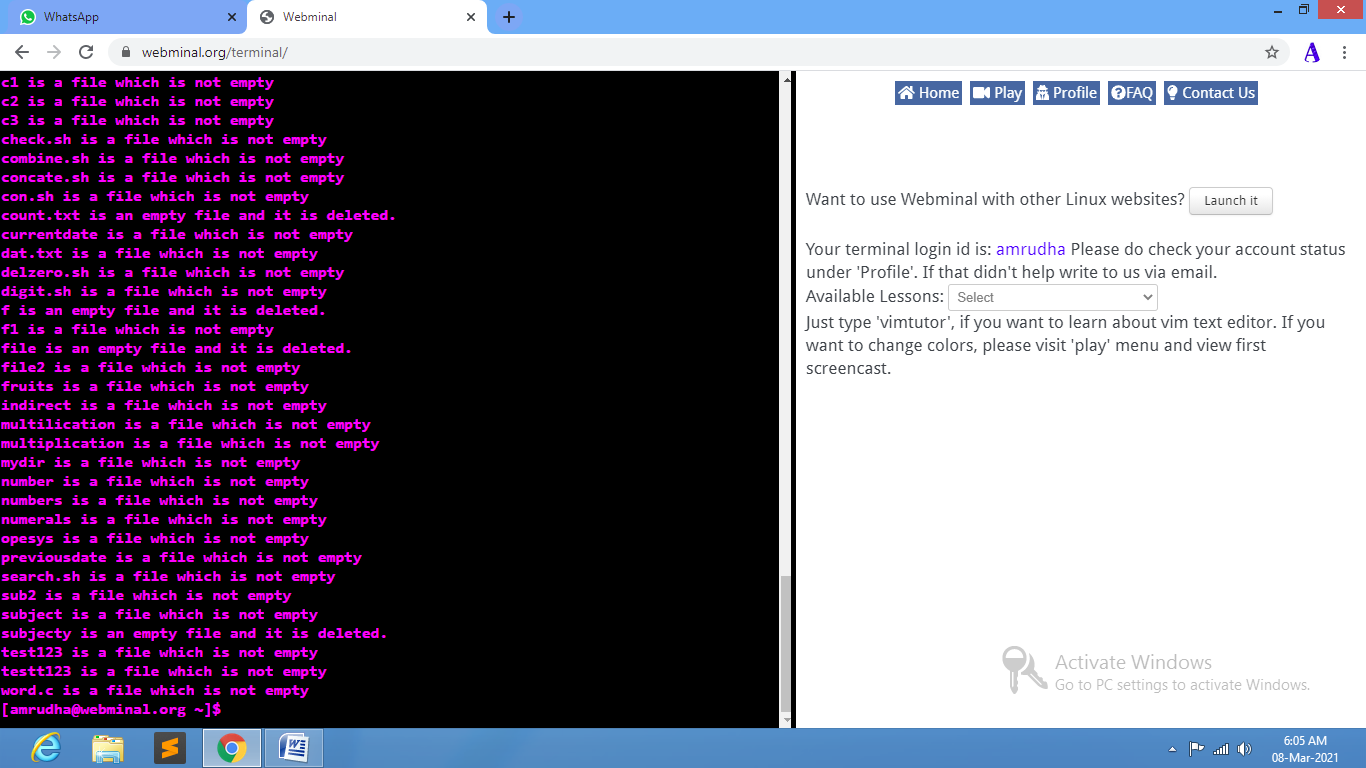
echo "$i is an empty file and it is deleted."

rm -rf $i

fi

done

**Output:**



**10.Write the shell program to reverse a number.**

**Algorithm:**

1.Get the given number n.  
2.Separate the digits of n individually.  
3.Then add the individual digit.  
4.Get the given number n.  
5.Separate the digits of n individually.  
6.Then add the individual digits and print the sum.  
7.Rearrange the digit in reverse order.

8.Stop

**Program:**

echo “enter the number”

read n

s=0

rev=0

while [ $n –gt 0 ]

do

s=$(( $n % 10))

rev=1 expr $rev \\* 10 + $s `

n=$(( $n / 10 ))

done

echo “reverse number of the entered digit is:$rev”

**Output:**

