

EX NO:2

SHELL PROGRAMMING

DATE:04.03.2021

Aim:

To write the following shell programs and execute in unix environment.

Programs:

1. Write a shell program for getting and displaying academic details. Inputs are name, roll no, three marks of students and outputs are name, rollno, total and average.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Using 'echo' command, print "name, rollno, m1, m2 and m3 respectively"
4. Using 'read' command, read name, rollno, m1, m2, m3 from the user
5. Using 'expr' command, get the total marks by adding m1, m2, m3 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 and average to the user.
7. Stop.

Program:

```
echo "name"
read name
echo $name
echo "rollno"
read rollno
echo $rollno
echo "m1"
read m1
echo $m1
echo "m2"
read m2
echo $m2
echo "m3"
read m3
echo $m3
total=`expr $m1 + $m2 + $m3`
echo $total
average=`expr $total / 3`
echo $average
```

Output:

```
[suuky@webminal.org ~]$vi suu.sh
[suuky@webminal.org ~]$sh suu.sh
name
lilly
lilly
rollno
29
29
mark1
90
90
mark2
90
90
mark3
90
90
total
270
avg
90
[suuky@webminal.org ~]$
```

2. Write a shell program to implement the arithmetic operations.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Using 'read' command, read the value of a and b from the user
4. Using 'expr' command, perform the arithmetic operations such as add, subtract, multiply and divide with a and b.
5. Using 'echo' command, print the addition, subtraction, multiplication, division results to the user.
6. Stop.

Program:

```
read a
read b
add=`expr $a + $b`
sub=`expr $a - $b`
mul=`expr $a \* $b`
div=`expr $a / $b`
echo "$add"
echo "$sub"
echo "$mul"
```

echo "\$div"

Output:

```
[suuky@webminal.org ~]$vi q.sh
[suuky@webminal.org ~]$sh q.sh
a
20
20
b
30
30
addition:
50
subtraction:
-10
multiplication:
600
division:
0
[suuky@webminal.org ~]$
```

3. Write a shell program to check whether the given number is positive, negative and zero.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Using 'read' command, read the value of a from the user
4. Using 'if' loop, check the conditions greater than zero for positive, less than for negative and zero.
5. Using 'echo' command, print whether the value is positive, negative or zero.
6. End the 'if' loop using fi.
7. Stop.

Program:

```
read a
if [ $a -gt 0 ]
then
echo " $a is positive"
elif [ $a -lt 0 ]
then
echo " $a is negative"
else
echo "$a is zero"
```

Output:

```
[suuky@webminal.org ~]$vi u.sh
[suuky@webminal.org ~]$sh u.sh
enter the number
20
the given no is positive
[suuky@webminal.org ~]$vi u.sh
[suuky@webminal.org ~]$sh u.sh
enter the number
-10
the given no is negative
[suuky@webminal.org ~]$vi u.sh
[suuky@webminal.org ~]$sh u.sh
enter the number
0
the given no is zero
[suuky@webminal.org ~]$
```

4. Write a shell program to form combinations for 1 2 3.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Using three 'for' loops ,specify the values for a,b and c in 1,2,3.
4. Using 'echo' command, print the values as combinations of the three numbers i.e 111 112 etc.,.
5. End the 'for' loops using done.
6. 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:

```
[suuky@webminal.org ~]$vi m.sh
[suuky@webminal.org ~]$sh m.sh
1 1 1
1 1 2
1 1 3
1 2 1
1 2 2
1 2 3
1 3 1
1 3 2
1 3 3
2 1 1
2 1 2
2 1 3
2 2 1
2 2 2
2 2 3
2 3 1
2 3 2
2 3 3
3 1 1
3 1 2
3 1 3
3 2 1
3 2 2
3 2 3
3 3 1
3 3 2
3 3 3
[suuky@webminal.org ~]$
```

5. Write a shell program to find the area of triangle, circle, square and rectangle using switchcase.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Using 'switch case' , evaluate the area of triangle, circle, square and rectangle.
4. Using 'echo' command, print "enter a value between 1 and 4".
5. Using the 'read' command, get 'a' value.
6. Inside the switch case write commands for the areas of triangle, circle, square and rectangle.
7. End 'case' using 'esac'.
8. Stop.

Program:

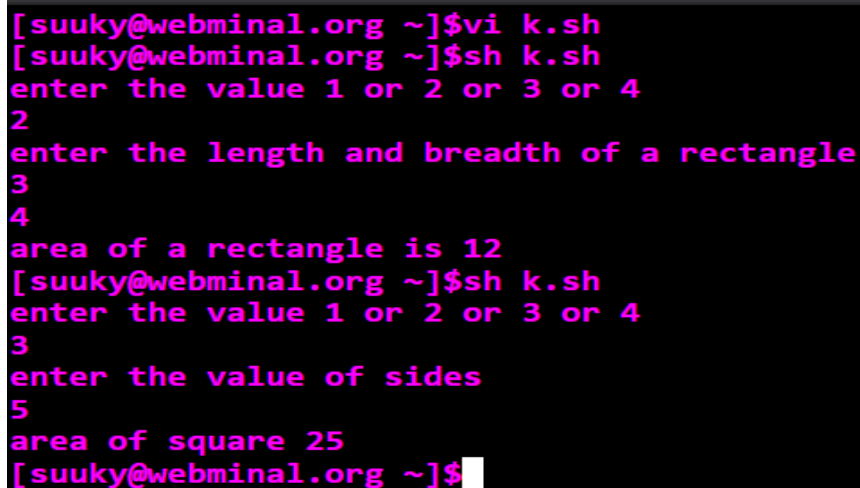
```
echo " enter a value between 1 and 4"
read a
case $a in
```

```

1)echo "area of triangle"
read b
read h
triangle=`expr $b \* $h \* 1 / 2`
echo "$triangle"
;;
2)echo "area of circle"
read r
circle=`expr $r \* $r \* 22 / 7`
echo "$circle"
;;
3)echo "area of square"
read c
square=`expr $c \* $c`
echo "$square"
;;
4)echo "area of rectangle"
read l
read b
rectangle=`expr $l \* $b`
echo "$rectangle"
;;
esac

```

Output:



```

[suuky@webminal.org ~]$vi k.sh
[suuky@webminal.org ~]$sh k.sh
enter the value 1 or 2 or 3 or 4
2
enter the length and breadth of a rectangle
3
4
area of a rectangle is 12
[suuky@webminal.org ~]$sh k.sh
enter the value 1 or 2 or 3 or 4
3
enter the value of sides
5
area of square 25
[suuky@webminal.org ~]$

```

6. Write a program to concatenate two strings and find the length of the resultant string.

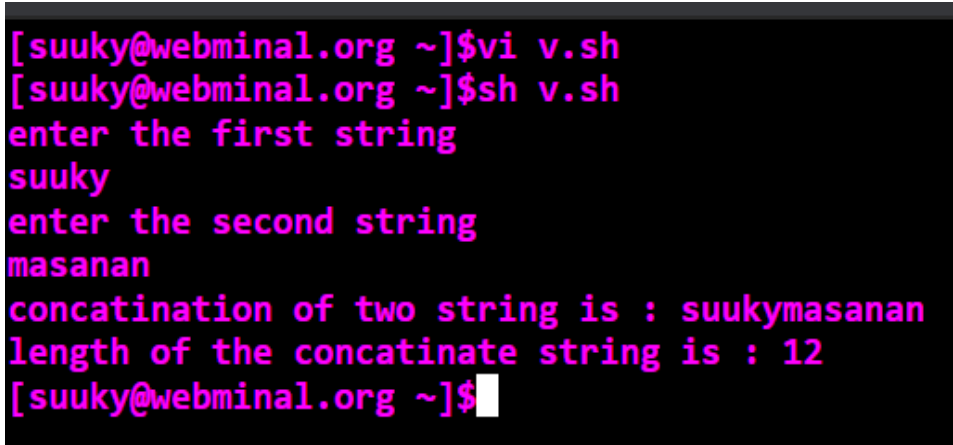
Algorithm:

- 1.Start
- 2.Create a file using vi command with filename.sh
- 3.Using 'read' command ,read two strings.
- 4.Concatenate two strings and store it in c.
- 5.Find the length of c by the code \${#c}
- 6.Using 'echo' command,display the result to the user.
- 7.End

Program:

```
read a
read b
c="${a}${b}"
echo "resultant string is $c"
echo "the length is ${#c}"
```

Output:



```
[suuky@webminal.org ~]$vi v.sh
[suuky@webminal.org ~]$sh v.sh
enter the first string
suuky
enter the second string
masanan
concatination of two string is : suukymasanan
length of the concatinated string is : 12
[suuky@webminal.org ~]$
```

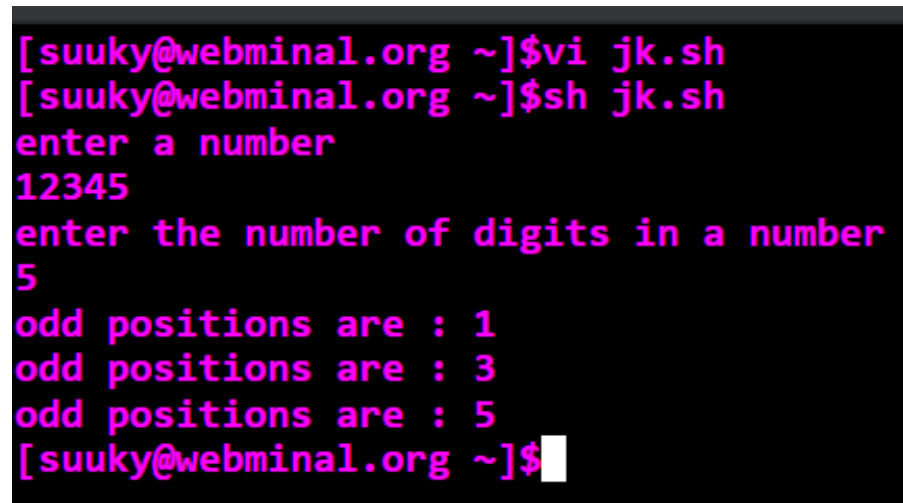
7. Write a program to display the digits which are in odd position in the given number.

Algorithm:

- 1.Start
- 2.Create a file using vi command with filename.sh
- 3.Using 'read' command, read n value
- 4.Initialize count =1 and find the length of the number entered by the user
- 5.Using 'while' and 'check' ,if count is less than or equal to length if yes,cut the number in that position and display it using echo and increment count by 2
- 6.Do it until the count is less than or equal to length of the input number
- 7.End

Program:

```
read n
count=1
len=${#n}
while [ $count -le $len ]
do
a=`echo $n | cut -c $count`
echo "$a"
count=`expr $count +2`
done
```

Output:

```
[suuky@webminal.org ~]$vi jk.sh
[suuky@webminal.org ~]$sh jk.sh
enter a number
12345
enter the number of digits in a number
5
odd positions are : 1
odd positions are : 3
odd positions are : 5
[suuky@webminal.org ~]$
```

8. Write a program to search an element in an array.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Initialize the array elements and initialise it to the user.

4. Using 'read' command, get the number to be searched and then set the flag as 0.
5. Use a 'for' loop and set conditional position as 0 and use 'if' loop inside it and check whether the position of the current number is equal to the position of the number entered, if yes change flag value to 1 or else flag remains 0.
6. Increment the position.
7. Display the result.
8. Stop.

Program:

```
a=(1 2 3 4 5)
echo "array elements are ${a[@]}"
echo "enter number to be searched "
read n
flag=0
for i in "${a[@]}"
do
if [ $i -eq $n ]
then
flag=1
fi
done
if [ $flag -eq 1 ]
then
echo "found"
else
echo "not found"
fi
```

Output:

```
[suuky@webminal.org ~]$vi rm.sh
[suuky@webminal.org ~]$sh rm.sh
Enter an array of string
lilly rose jasmin floral fauna
Enter the search element in an array
rose
The search element  rose  is FOUND
[suuky@webminal.org ~]$
```

9. Write a program to delete the zero sized file using if and for.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Using 'echo' and 'read' command, get the filename from the user.
4. Using 'if' loop, check whether the file exists or has filesize greater than 0 or else file is deleted using 'rm' command.
5. Displays file not exists, otherwise.
6. Stop.

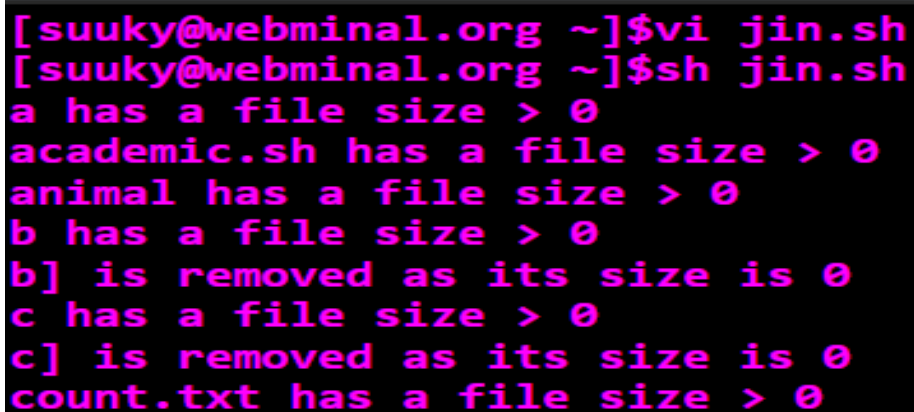
Program:

```
echo "enter filename"
read fn
if [ -e $fn ]
then
echo "file exists"
if [ -s $fn ]
then
echo "filesize > 0"
else
echo "empty file deleted"
```

M.SUUKY
19CSE029

```
rm $fn
fi
else
echo "file not exists"
fi
```

Output:



```
[suuky@webminal.org ~]$vi jin.sh
[suuky@webminal.org ~]$sh jin.sh
a has a file size > 0
academic.sh has a file size > 0
animal has a file size > 0
b has a file size > 0
b] is removed as its size is 0
c has a file size > 0
c] is removed as its size is 0
count.txt has a file size > 0
```

10. Write a program to reverse a number.

Algorithm:

1. Start
2. Create a file using vi command with filename.sh
3. Using 'read' command, get the a value from the user and assume r and re values as 0 and null respectively.
4. Using 'while' loop, check if a value is not equal to zero, then perform modulus and division operations, to get the reversed number.
5. Displays reversed number.
6. Stop.

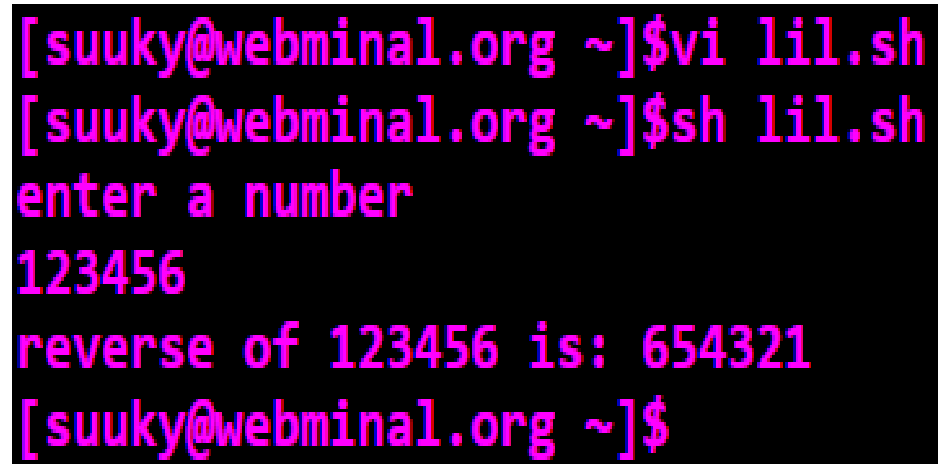
Program:

```
read a
r=0
re=""
```

M.SUUKY
19CSE029

```
while [ $a ne 0 ]
do
r=`expr $a % 10`
a=`expr $a / 10`
re="$re${r}"
done
echo "reversed number is ${re}"
```

Output:



```
[suuky@webminal.org ~]$vi lil.sh
[suuky@webminal.org ~]$sh lil.sh
enter a number
123456
reverse of 123456 is: 654321
[suuky@webminal.org ~]$
```

Observation(20)	
Record(5)	
Total(25)	
Initial	

Result:

Thus the shell commands were executed and outputs were noted.

