Ex.No: 02 **Shell Programming**
Date : 04.03.2021

Aim:

To understand the shell programming and execute programs

1.Getting and displaying academic details

Algorithm:

- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter name,roll number,mark1,mark2 and mark3 through the keyword echo and scan them all through the keyword read
- Calculate total and average for the three marks
- Display the name,roll number,three marks,total and average.
- End program

Program:

```
echo “enter the name”
read name
echo “enter your roll number”
read roll
echo “enter the mark1”
read m1
echo “enter the mark2”
read m2
echo “enter the mark3”
read m3
echo “name::$name”
echo “roll number::$roll”
tot=` expr $m1+$m2+$m3 `
echo “total::$tot”
avrg=` expr $tot / 3 `
echo “average::$avrg”
```


K . Aafreen Benazir
19cse001

Output:

```
[AafreenK@webminal.org ~]$sh academ.sh
enter your name
aaf
enter your roll number
1
enter the mark1
90
enter the mark2
90
enter the mark3
90
name :: aaf
roll number :: 1
marks are :: mark 1 - 90,mark2 - 90,mark3 - 90
total :: 270
average :: 90
```

2.Executing arithmetic operation

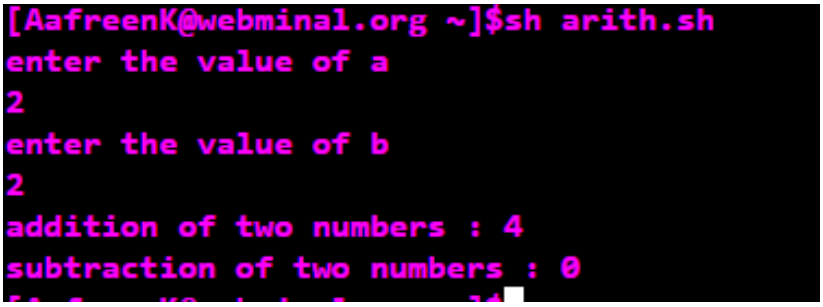
Algorithm:

- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter a value through the keyword echo and scan them all through the keyword read
- Perform the addition,subtraction operation
- Display the result
- End program

Program:

```
echo “enter the value of a”
read a
echo “enter the value of b”
read b
c=a+b
echo “addition of two numbers :: $c”
d=a-b
echo “subtraction of two numbers :: $d”
```

Output:



3.Check whether a number is positive negative or zero

Algorithm:

- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter a number through the keyword echo and scan it through the keyword read
- Check whether the given number is positive or negative or equal to zero using if ,elif and else.
- Display the result
- End program

Program:

```
echo “enter a number”
read a
if [ $a –eq0 ]
then
 echo “zero”
elif [ $a –lt0 ]
then
 echo “negative”
else
 echo “positive”
fi
```

Output:

```
[AafreenK@webminal.org ~]$vi pnz.sh
[AafreenK@webminal.org ~]$sh pnz.sh
enter a number
2
positive
```

4.Find the combinations of numbers 1 2 3

Algorithm:

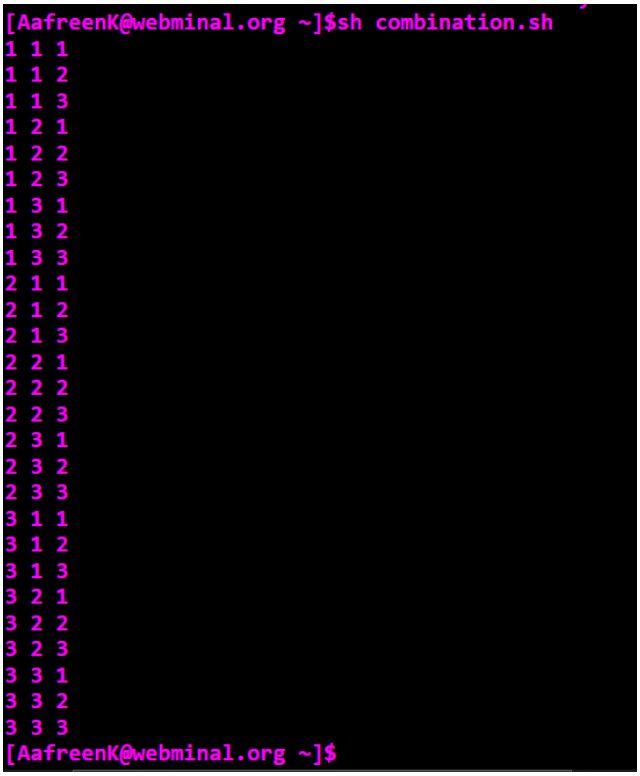
- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Use for loop and iterate through the values 1 2 3 with variable i
- Use nested loop iterate through the values 1 2 3 with variable j
- Use another nested loop iterate through the values 1 2 3 with variable k
- Print i , j and k
- End for loops

Program :

```
for i in 1 2 3
do
 for j in 1 2 3
 do
   for k in 1 2 3
   do
      echo $i $j $k
    done
   done
done
```


K . Aafreen Benazir
19cse001

Output:



5. find the area of the square , circle and rectangle using switch case

Algorithm:

- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter a side or radius
- List options 1.square 2.Circle 3.Rectangle
- Prompt user to choose a option
- Use Switch case to compute result based on options
- Option 1: Area=side*side for square
- Option 2: Area = 22/7 * radius * radius
- Option 3: Area = length * breadth
- End switch case
- End program

Program:

```
echo “Enter a side or radius : ”
read a
echo “1.square 2.circle 3.rectangle”
echo “Choose option : ”
read c
case $c in
 1)
   Area=$(($a * $a))
   echo “Area of the square is ${Area}”
   ;;
2)
   Area=$(($a * $a))
   Area=$(($Area * 22))
   Area=$(($Area /7))
   echo “Area of the circle is ${Area}”
   ;;
3)
   echo “Enter the breadth : ”
   read b
   Area=$(($a * $b))
   echo “Area of a rectangle is ${Area}”
   ;;
esac
```

Output:

```
[AafreenK@webminal.org ~]$sh switch.sh
enter a side or radius
4
1.square 2.circle 3.rectangle
enter option :
1
Area of a square is 16
```


K . Aafreen Benazir
19cse001

6.Concatenate two strings and find the length of the resultant string

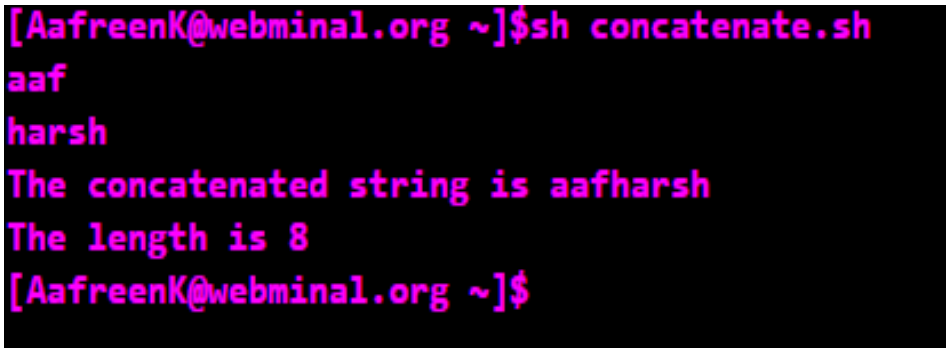
Algorithm:

- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter two strings through the keyword echo and scan it through the keyword read
- Concatenate two strings and store it in a variable c
- Find the length of c by the code ${#c}
- Display the result
- End program

Program:

```
echo “enter first string”
read a
echo “enter second string”
read b
c=”${a}${b}”
echo “concatenation of two strings $c”
echo “the length of the resultant string ${#c}”
```

Output:



7.Display the digits which are in odd position in the given number

Algorithm:

- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter a number through the keyword echo and scan it through the keyword read
- Initialize a variable count =1 and find the the length of the number that the user has entered
- Use 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
- Do it until the count is less than or equal to length of the input number
- End program

Program:

```
echo “enter a number”
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:

8.Search a element in a array

Algorithm:

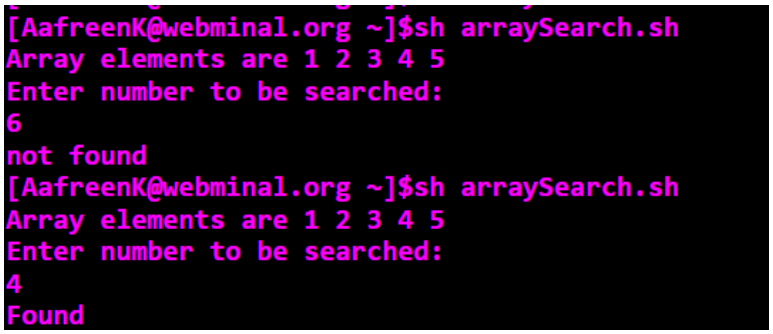
- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Initialize the array elements and display it to the user
- Prompt the user to enter a number to be searchedthrough the keyword echo and scan it through the keyword read
- Set flag=0 and start a for loop with condition position=0 and start a if loop inside the for loop and check whether the element in the current position is equal to the element to be searched if yes set flag=1 else flag=0
- Increment the position
- Display the result
- End program

Program:

```
array1=(1 2 3 4 5)
echo “Array elements are ${array1[@]}”
echo ”enter number to be searched : “
read n
flag=0
for i in “${array1[@]}”
do
  if [ $i -eq $n ]
  then
    flag=1
  fi
done
if [ $flag -eq1 ]
then
   echo “Found”
else
   echo “Not found”
```


K . Aafreen Benazir
19cse001

Output:



9. Delete a zero sized file

Algorithm:

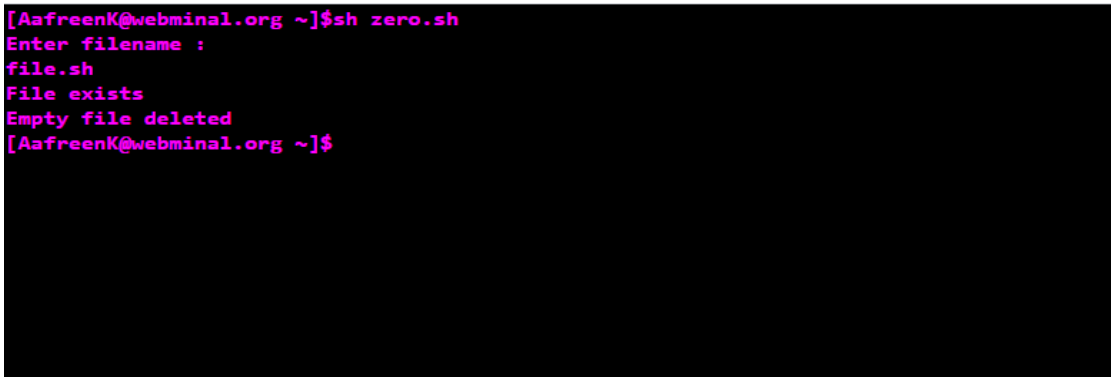
- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter a file name to be deleted through the keyword echo and scan it through the keyword read
- Check if the file exists in the directory. If it exists,then prompt to the user that the file exists through keyword echo.If not display a message that the file does not exists
- Check for the size of the file.If the file size is equal to 0, delete the file through the keyword 'rm $filename'.If the file size is not equal to 0,display a message to the user that the file size exceeds zero
- End program

Program:

```
echo "Enter filename :"
read filen
if [ -e $filen ]
then
  echo "File exists"
  if [ -s $filen ]
  then
    echo "File has size .0"
  else
    echo "Empty file deleted"
    rm $filen
```

```
  fi
else
 echo “file not exists”
fi
```

Output:



10.Reverse a number

Algorithm:

- Create a file with extension .sh and enter into vi(virtual editor).
- Press i to enter into insert mode
- Prompt the user to enter a number through the keyword echo and scan it through the keyword read
- Initialize reverse as 0
- Check if the number equals to 0 in a while loop.When this condition fails,compute the result of the ‘number mod 10’ to the variable r.divide the number by 10 and assign the value to the variable n.Finally assign the concatenated value of the variables rev and r to rev.
- Continue the iteration of the loop until the number becomes 0
- Display the reversed number to the user through the keyword echo
- End program

Program:

```
echo “Enter a number :”
read n
r=0
rev=” ”
while [ $n –ne 0 ]
do
  r=` expr $n % 10`
  n=` expr $n / 10`
  rev=”${rev}${r}”
done
echo “The reverse number is ${rev}”
```

Output :

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

Result:

Thus ,the Shell programs were executed and outputs were noted.