Shellprogram - Shell programming lab list programs

Bachelor's of computer Application (BCA) (University of Calicut)

PART B – SHELL Programming

1. Write a shell script to find area of a circle

```
echo "Enter the radius of the circle"
read r
echo "3.14*$r*$r" | bc
area=$(echo "3.14*$r*$r" | bc)
echo "a=$area"
```

2. Write a shell script to find given number is even or odd

```
echo "Enter number"
read number
even_odd=$((number%2))

if test $even_odd -eq 0
then
        echo "even"
else
        echo "Odd"
fi
```

3. Write a shell script to make a menu driven calculator using case

```
while true
do

echo "Enter operator"
```

```
echo "1. for addition"
echo "2. for subtraction"
echo "3. for multiplication"
echo "4. for division"
echo "5. exit"
read op;

case $op in
    1)
        echo "Enter first number"
        read n1
        echo "Enter second number"
        read n2
        res=$((n1+n2))
        operator="addition"
        ;;
    2)
        echo "Enter first number"
         read n1
        echo "Enter second number"
        read n2
        res=$((n1-n2))
        operator="subtraction"
        ;;

    3)
        echo "Enter first number"
        read n1
        echo "Enter second number"
        read n2
        res=$((n1*n2))
```

```
        operator="multiplication"
        ;;
    4)
        echo "Enter first number"
        read n1
        echo "Enter second number"
        read n2
         res=$((n1/n2))
        operator="division"
        ;;
    5)   exit
        ;;
    *)   echo "Enter proper operator"
        operator="Invalid Operation"
        res="null"
        ;;
esac
echo "Result after $operator is $res"
done
```

4. Write a shell script to find the greatest of three numbers

```
echo "Enter first number"
read n1
echo "Enter second number"
read n2
echo "Enter third number"
read n3

if test $n1 -gt $n2
```

```
then
    if test $n1 -gt $n3
    then
        echo "$n1 is the greatest"
    else
        echo "$n3 is the greatest"
    fi
elif test $n2 -gt $n3
then
    echo "$n2 is greatest"
else
    echo "$n3 is greatest"
fi
```

5. Write a shell script to compute mean and standard deviation of three numbers

```
echo "Enter first number"
read n1
echo "Enter second number"
read n2
echo "Enter third number"
read n3
mean=$(echo "scale=2;($n1+$n2+$n3)/3" | bc  )
n1_mean=$(echo "scale=2;$n1-$mean" | bc )
echo "diff1: $n1_mean"
squared_diff_n1=$(echo "scale=2;$n1_mean^2" | bc)
echo "diff1^2: $squared_diff_n1"



n2_mean=$(echo "scale=2;$n2-$mean" | bc )
echo "diff2: $n2_mean"
```

```
squared_diff_n2=$(echo "scale=2;$n2_mean^2" | bc)
echo "diff2^2: $squared_diff_n2"

n3_mean=$(echo "scale=2;$n3-$mean" | bc )
echo "diff3: $n3_mean"
squared_diff_n3=$(echo "scale=2;$n3_mean^2" | bc)
echo "diff3^2: $squared_diff_n3"

sum_squared_diff=$(echo "scale=2;$squared_diff_n1+$squared_diff_n2+$squared_diff_n3" | bc)

echo "sum of squared differencences: $sum_squared_diff"
mean_squared_diff=$(echo "scale=2;$sum_squared_diff/2" | bc)
echo "Mean of squared differences: $mean_squared_diff"

stand_dev=$(echo "scale=2;sqrt($mean_squared_diff)"|bc)

echo "Mean=$mean"
echo "Standard Deviation=$stand_dev"
```

6. Write a shell script to find sum of all digits from a given number

```
echo "Enter number"
read number

sum=0
while test $number -ne 0
do
    digit=$((number%10))
    sum=$((sum+digit))
    number=$((number/10))
```

```
done
echo "Sum of digits: $sum"
```

7. Write a shell script to find reverse of a number

```
echo "Enter number"
read number
reverse=""
sum=0
while test $number -ne 0
do
    digit=$((number%10))
    reverse="$reverse$digit"
    number=$((number/10))
done
echo "$reverse"
```

8. Write a shell script to find prime numbers up to a given number

```
echo "Enter limit"
read limit

start=2
while test $start -le $limit
do
    num=$start
    flag=0
    i=2
    k=$((num/2))
```

```
        while test $i -le $k
        do
                rem=$((num%i))
                        if test $rem -eq 0
                        then
                                flag=1
                                break
                        fi
                i=$((i+1))
        done
        if test $flag -eq 0
        then
                echo "$num is Prime"
        fi
        start=$((start+1))

done
```

9. Write a shell script to find N fibonacii numbers

```
echo "Enter limit"
read limit

start1=0
start2=1
count=3
echo "$start1"
echo "$start2"

while test $count -le $limit
do
        fib=$((start1+start2))
```

```
        echo $fib
        start1=$start2
        start2=$fib
        count=$((count+1))

done
```

10. Write a shell script to check whether a given number is Armstrong or not

```
echo "Enter number"
read number
number_cpy=$number
sum=0
count=0;
while test $number -ne 0
do
    number=$((number/10))
    count=$((count+1))
done
echo "Count of digits:$count"
sum=0
number=$number_cpy
while test $number -ne 0
do
        digit=$((number%10))
        powered_digit=$((digit**count))
        sum=$((sum+powered_digit))
        echo "Digit:$digit Powered Digit:$powered_digit"
        number=$((number/10))
```

```
done
echo "sum of powered digits:$sum"
if test $sum -eq $number_cpy
then
        echo "-------------------Armstrong--------------------"
else
        echo "----------------Not Armstrong--------------------"
fi
```

11. Write a shell script to reverse a string and check whether a given string is palindrome or not

```
echo "Enter a string"
read  string
reverse=$(echo $string | rev)
echo "Reverse of $string is $reverse"

if test $string = $reverse
then
      echo "Palindrome"
else
      echo "Not Palindrome"
fi
```

12. Write a shell script to count no of line, words and characters of a input file

```
echo Enter the filename
read f
words=$(cat $f | wc -w)
echo "words:$words"
```

```
characters=$(cat $f|wc -c)
echo "Number of characters in $file is $characters"

lines=$(cat $f|wc -l)
echo "No of lines is $lines"
```

13.Write a shell script to convert all the contents into the uppercase in a particular file in Unix

```
echo "Enter file name"
read fname
tr '[:lower:]' '[:upper:]' < $fname > output.txt
cat output.txt
```

14. Write a shell script to find the value of one number raised to the power of another. Two numbers are entered through the keyboard

```
echo "Enter base"
read base
echo "Enter exponent"
read exponent

power=$(echo "$base^$exponent" | bc)
echo "$power"
```

15. Write a shell script find the factorial of a given number

```
echo "enter no"
read fact
```

```
fact1=$fact
while test $fact1 -ne 1
do
        fact=$((fact * (fact1-1)))
        fact1=$((fact1-1))

done
echo "fact=$fact";
```

16. An employee Basic Pay is input through keyboard where DA is 40% of basic pay and HRA is 20% of basic pay. Write a shell script to calculate gross salary, Gross Salary =Basic Pay + DA + HRA

```
echo "Enter basic"
read basic
da=$(echo "scale=2;0.4*$basic" | bc)
echo "DA=$da"

hra=$(echo "scale=2;0.2*$basic" | bc)
echo "HRA=$hra"

gross=$(echo "scale=2;$basic+$da+$hra" | bc)


echo "Gross salary=$gross"
```

17. Write a shell script to find the average of the numbers entered as command line arguments

```
echo "First argument:$1"
echo "Second argument:$2"
```

```
echo "Third argument:$3"



average=$(echo "scale=2;($1+$2+$3)/3" | bc)
echo "AVerage of first 3 numbers:$average"

sum=0
i=0
for number in "$@"
do
    sum=$(echo "scale=2;$sum+$number" | bc)
    i=$((i + 1));
done

avg1=$(echo "scale=2;$sum/$#" | bc)
avg2=$(echo "scale=2;$sum/$i" | bc)

echo "Average of n numbers:$avg1"
echo "Average of n numbers:$avg2"
```

18. Write a shell script which whenever gets executed displays the message Good Morning/Good afternoon /Good Evening depending on the time it gets executed

```
check=$(date +%H)
echo $check
if [ $check -ge 06 -a $check -le 12 ]
then
        echo "Good morning"
elif [ $check -ge 12 -a $check -le 17 ]
then
```

```
    echo "Good afternoon"
else
    echo "Good evening"
fi
```

19. Write a shell script to Display Banner, calendar of given year

```
echo "Enter a maxiumum of 10 letters"

read text

banner $text

echo "Enter year"

read year

cal $year
```

20 Write a shell script to display current date and time, number of users, terminal name, login date and time

```
current_date=`date`
echo "Today's date: $current_date"

no_of_users=`who|wc -l`

echo "No of users: $no_of_users"

who
```

21. Write a shell script which uses all the file test operators

```
FILE=$1

# The -e operator tests for file existence
if [ -e $FILE ]
then
  # If the operator returns true, print a message saying the file exists.
  echo "$FILE exists"
else
  # If the operator returns true, print a message saying the file doesn't exist,
    # then creates the file with the name you defined in the FILE variable.
  #echo "$FILE does not exist, creating new file" && touch test.txt"
  echo "$FILE does not exist"
fi

if [ -f $FILE ]
then
          echo "$FILE is a regular file"



else
  echo "$FILE not a regular file"
fi

if [ -s $FILE ]
then
       echo "$FILE is not empty"
```

```
else
  echo "$FILE empty"
fi

if [ -r $FILE ]
then
        echo "$FILE has read permissiom"


else
  echo "$FILE does not have read permission"
fi

if [ -w $FILE ]
then
        echo "$FILE has write permissiom"


else
  echo "$FILE does not have write permission"
fi
if [ -x $FILE ]
then
        echo "$FILE has executable permissiom"


else
  echo "$FILE does not have executable permission"
fi

if [ -O $FILE ]
```

```
then
        echo "You are the owner of $FILE"


else
  echo "You are not the owner of $FILE"
fi

if [ -N $FILE ]
then
        echo "$FILE modified since last read"


else
  echo "not modified since last read"
fi
if [ -O $FILE ]
then
        echo "You are the owner of $FILE"


else
  echo "You are not the owner of $FILE"
fi

echo "Enter file1"
read file1

echo "Enter file2"
read file2
```

```
if [ $file1 -nt $file2 ]
then
        echo "$file1 is newer than $file2"



else
  echo "$file1 is not newer than $file2"
fi
```

22. Write a shell script to copy the contents of file to another. Input file names through command line. The copy should not be allowed if second file exists.

```
source=$1
destination=$2

if [ -e $destination ]
then
        echo "$destination exists. contents will be overwritten"
else
        cp $source $destination
fi
```

23. Write a shell script to find number of vowels, consonants, numbers in a given string.

```
echo "Type any String"
  read string

  length=`echo $string | wc -c`
```

```
echo $length
nvowels=0
nconsonants=0
ndigits=0

while [ $length -gt 1 ]
do
   length=`expr $length - 1`
   h=`echo $string | cut -c $length`

   case $h in
   [AaEeIiOoUu]) nvowels=`expr $nvowels + 1`
   ;;

   [BbCcDdFfGgHhJjKkLlMmNnPpQqRrSsTtVvWwXxYyZz])
   nconsonants=`expr $nconsonants + 1`
   ;;

   [0-9]) ndigits=`expr $ndigits + 1`
   ;;
esac
done

echo "Number of Vowels      : $nvowels"
echo "Number of Consonants  : $nconsonants"
echo "Number of Digits      : $ndigits"
```

24. Write a shell script to perform operations like display, list, make directory and copy, rename, delete

```
while true
do

echo "Enter operations
 1. Display contents of a file
 2. copy a file to another file
 3. Rename a file
 4. delte a file
 5. List the contents in a directory
 6. make directory "

read choice

case $choice in
        1)
                echo "DIsplaying contents of a file. Please enter the name of the file"
                read filename
                cat $filename
                ;;
        2)
                echo "Copying a file to another"
                echo "Enter source"
                read source
                echo "Enter destination"
                read destination

                cp $source $destination
                ;;
        3)
                echo " Enter the name of the file to be renamed"
                read filename
```

```
        echo "Enter the new name"
        read newfilename

        mv $filename $newfilename
        ;;
    4)
        echo "Enter the name of the file to be deleted"
        read filename
        rm $filename
        ;;
    5)
        echo " Enter the name of the directory to be listed"
        read dirname
        ls $dirname
        ;;
    6)
        echo "Enter the name of the directory to be created"
        read dirname
        mkdir $dirname
        ;;
esac

done
```

25. Write a shell script to compare two files and remove one of them if they are same

```
echo "Enter file1"
read file1
```

```
echo "Enter file2"
read file2

count=$(diff $file1 $file2 | wc -l)
echo "words:$words"
if test $count -eq 0
then
        echo "They are same. So removing $file2"
    rm $file2
else
    echo "THey are different. Preserving both of them"
fi
```