Shell programs

1. Perform Arithmetic Operations

#/bin/sh

a=1

b=2

sum=` expr $a + $b `

echo $sum   
3

1. #/bin/sh

a=12;b=32

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

echo $res

384

1. #/bin/sh

x=120

y=17

ans=` expr $x / $y `

echo $ans

7

1. #/bin/sh

x=120

y=17

sub=` expr $x - $y `

echo $sub

103

1. #/bin/sh

Name=”James Bond”

echo $Name

James Bond

1. #/bin/sh

Num=123

Echo ` expr $Num + 3`

126

1. #/bin/sh

msg1=Hello

msg2=Welcome!

echo $msg1 $msg2

Hello Welcome!

1. #/bin/sh

var1="HI"

var2="There"

echo $var1 $var2

HI There

1. #/bin/sh

Echo ‘”welcome world”’ or `”welcome world”` 🡪 command not found..

1. #/bin/sh

echo $(( 10 +20 +30))

60

1. #/bin/sh

echo " i have \$100"

echo " i have $100"

echo ' i have $100'

echo printing date and time `date`

echo $?

i have $100

i have 00

i have $100

printing date and time Fri Apr 20 10:36:23 IST 2018

0

Find the error in the below program and correct it

========================================

1. #/bin/sh

myname=Vivek

myos =TroubleOS

myno=5

echo "My name is $myname"

echo "My os is $myos"

echo "My number is myno, can you see this number" – Add $myno

My name is Vivek

My os is TroubleOS

My number is myno

13 . #/bin/sh

echo enter ur name

read name

echo "hello $name"

enter ur name

Gaurav

hello Gaurav

14 . #/bin/sh

a=0

while [ "$a" -lt 10 ]

do

b="$a"

while [ "$b" -ge 0 ]

do

echo -n "$b "

b=` expr $b - 1 `

done

echo

a=` expr $a + 1 `

done

0

1 0

2 1 0

3 2 1 0

4 3 2 1 0

5 4 3 2 1 0

6 5 4 3 2 1 0

7 6 5 4 3 2 1 0

8 7 6 5 4 3 2 1 0

9 8 7 6 5 4 3 2 1 0

15 . #/bin/sh

a=0

while [ $a -le 10 ]

do

echo $a

a=`expr $a + 1`

done

0

1

2

3

4

5

6

7

8

9

10

16 . Example for infinit loop

=====================

#/bin/sh

a=10

while [ $a -ge 10 ]

do

echo $a

a=`expr $a + 1`

done

0 till infinite

infinite loop

16 . #/bin/sh

a=0

while [ $a -le 10 ]

do

if [ $a -eq 5 ]

then

break

fi

echo $a

a=`expr $a + 1`

done

0

1

2

3

4

17. for var1 in 1 2 3

do

for var2 in 0 5

do

if [ $var1 -eq 2 -a $var2 -eq 0 ]

then

break 2

else

echo "$var1 $var2"

fi

done

done

1. 0

1 5

18. #/bin/sh

for var1 in 1 2 3

do

echo $var1

done

1

2

3

19. NUMS="1 2 3 4 5 6 7"

for NUM in $NUMS

do

Q=`expr $NUM % 2`

if [ $Q -eq 0 ]

then

echo "Number is an even number!!"

continue

fi

echo "Found odd number"

done

Found odd number

Number is an even number!!

Found odd number

Number is an even number!!

Found odd number

Number is an even number!!

Found odd number

20. #/bin/sh

# Usage of test command

if test $1 -gt 0

then

echo "$1 number is positive"

else

echo "$1 is negative"

fi

is negative

21. #/bin/sh

#To print the date ,No of users logged in and the uptime of the server

Date=`date`

echo "Date is $Date"

Users=`who | wc -l`

echo "Logged in Users :$Users"

UP=`date ;uptime`

echo "uptime is $UP"

Date is Fri Apr 20 11:36:55 IST 2018

Logged in Users :23

uptime is Fri Apr 20 11:36:55 IST 2018

11:36:55 up 62 days, 7:52, 23 users, load average: 0.19, 0.06, 0.01

22. #/bin/sh

echo "select Department"

echo -e "1. Technical \t 2. Developing

echo "Select your choice [1 or 2]

read depart

if test $depart -eq 1

then

echo "You had selected technical as your department"

else

if test $depart -eq 2

then

echo "you have selected developing as your department"

else

echo "No valid Department found"

fi

fi

select Department

1. Technical 2. Developing

echo Select your choice [1 or 2]

1

You had selected technical as your department

21. if cat $1

then

echo -e "\n\n File $1, found and successfully echoed "

fi

abc

abc

abc

abc

def

def

22. #/bin/sh

if rm $1

then

echo "$1 file deleted"

fi

23. #/bin/sh

if [ $# -eq 0 ]

then

echo "$0 - Then you should supply one integer"

exit 1

fi

30.sh - Then you should supply one integer

24 . #/bin/sh

if [ -z $1 ]

then

course="\*\*\*Exam Date not scheduled\*\*\*"

elif [ -n $1 ]

then

course=$1

fi

case $course in

"Bcom")echo "Exam date for $course is 15th November";;

"Mcom")echo "Exam date for $course is 20th November";;

"BCA")echo " Exam date for $course is 18th November";;

"MCA")echo " Exam date for $course is 22nd November";;

\*)echo "Sorry check your choice $course";;

esac

Sorry check your choice \*\*\*Exam Date not scheduled\*\*\*

25.

if test $1 -eq 1 && test $2 == JAN

then

echo "it is equal"

elif test $1 -eq 1 || test $2 == JAN

then

echo "it is not equal"

else

echo "unknown value"

fi

35.sh: line 2: test: -eq: unary operator expected

35.sh: line 5: test: -eq: unary operator expected

35.sh: line 5: test: ==: unary operator expected

unknown value

26. #/bin/sh

if [ $1 -eq 1 ] && [ $2 == JAN ]

then

echo "it is equal"

elif [ $1 -eq 1 ] || [ $2 == JAN ]

then

echo "it is not equal"

else

echo "unknown value"

fi

36.sh: line 2: [: -eq: unary operator expected

36.sh: line 5: [: -eq: unary operator expected

36.sh: line 5: [: ==: unary operator expected

unknown value

27. for var1 in 0 2 4 6 8

do

echo "$var1"

done

0

2

4

6

8

28. #/bin/sh

for var1 in 0 2 4 6 8

do

for var2 in 1 3 5

do

echo "$var1 $var2"

done

done

0 1

0 3

0 5

2 1

2 3

2 5

4 1

4 3

4 5

6 1

6 3

6 5

8 1

8 3

8 5

29. if [ $# -eq 0 ]

then

echo "Error - Number missing form command line argument"

echo "Syntax : $0 number"

echo "Use to print multiplication table for given number"

exit 1

fi

n=$1

for i in 1 2 3 4 5 6 7 8 9 10

do

echo "$n \* $i = ` expr $i \\* $n `"

done

Error - Number missing form command line argument

Syntax : 39.sh number

Use to print multiplication table for given number

30 . #/bin/sh

for (( i = 0 ; i <= 5; i++ ))

do

echo "Welcome $i times"

done

Welcome 0 times

Welcome 1 times

Welcome 2 times

Welcome 3 times

Welcome 4 times

Welcome 5 times

31. #/bin/sh

if [ $1 -gt 0 ]; then

echo "$1 is positive"

elif [ $1 -lt 0 ]

then

echo "$ is negative"

elif [ $1 -eq 0 ]

then

echo "$1 is zero"

else

echo "oops! $1 is not number ,give number"

fi

41.sh: line 2: [: -gt: unary operator expected

41.sh: line 4: [: -lt: unary operator expected

41.sh: line 7: [: -eq: unary operator expected

oops! is not number ,give number

32. for (( i = 1; i <= 5; i++ ))      ### Outer for loop ###  
do  
  
    for (( j = 1 ; j <= 5; j++ )) ### Inner for loop ###  
    do  
          echo -n "$i "  
    done  
  
  echo "" #### print the new line ###  
  
done

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

5 5 5 5 5

33. write a script that prints table of any number.

#!/bin/bash

echo “Enter The Number upto which you want to Print Table: ”

read n

i=1

while [ $i -ne 10 ]

do

i=$(expr $i + 1)

table=$(expr $i \\* $n)

echo $table

done

“Enter The Number upto which you want to Print Table: ”

5

10

15

20

25

30

35

40

45

50

34. find if the number is odd or even

#!/bin/bash

echo "Enter The Number"

read n

num=$(expr $n % 2)

if [ $num -eq 0 ]

then

echo "is a Even Number"

else

echo "is a Odd Number"

fi

Enter The Number

5

is a Odd Number

35. to find the Factorial

#!/bin/bash

echo "Enter The Number"

read a

fact=1

while [ $a -ne 0 ]

do

fact=$(expr $fact \\* $a)

a=$(expr $a - 1)

done

echo $fact

Enter The Number

4

24

36. To find the Armstrong number

#!/bin/bash

echo "Enter A Number"

read n

arm=0

temp=$n

while [ $n -ne 0 ]

do

r=$(expr $n % 10)

arm=$(expr $arm + $r \\* $r \\* $r)

n=$(expr $n / 10)

done

echo $arm

if [ $arm -eq $temp ]

then

echo "Armstrong"

else

echo "Not Armstrong"

fi

Enter A Number

8

512

Not Armstrong

37. To check whether the number is prime or not

#!/bin/bash

echo “Enter Any Number”

read n

i=1

c=1

while [ $i -le $n ]

do

i=$(expr $i + 1)

r=$(expr $n % $i)

if [ $r -eq 0 ]

then

c=$(expr $c + 1)

fi

done

if [ $c -eq 2 ]

then

echo “Prime”

else

echo “Not Prime”

fi

“Enter Any Number”

65

“Not Prime”

-bash-4.1$ sh 47.sh

“Enter Any Number”

61

“Prime”

38.

#!/bin/sh

echo "File Name:$0"

echo "First Parameter:$1"

echo "First Parameter;$2"

echo "Quoted Values:$@"

echo "Quoted Values:$\*"

echo "Total Number of Parameters:$#"

File Name:53.sh

First Parameter:

First Parameter;

Quoted Values:

Quoted Values:

Total Number of Parameters:0

39.

#!/bin/sh

math(){

#local variable x and y with passed args

local x=$1

local y=$2

echo $(( $x + $y ))

}

echo "x:$x and y:$y"

math 12 5

x: and y:

17

40. Usage of keyword function

#!/bin/sh

function display(){

echo Enter first no:

read fno

echo Enter second no:

read sno

tno=`expr $fno + $sno`

echo $tno

}

display

Enter first no:

20

Enter second no:

30

50

41.

#!/bin/sh

myfunc(){

echo "I was called as : $@"

x=2

}

###Main script starts here

echo "Script was called with $@"

x=1

echo "x is $x"

myfunc 1 2 3

echo "x is $x"

Script was called with

x is 1

I was called as : 1 2 3

x is 2

42.

#!/bin/sh

myfunc(){

echo "I was called as : $@"

echo "the no of arguments passed : $#"

echo "$\*"

x=2

}

###Main script starts here

echo "Script was called with $@"

x=1

echo "x is $x"

myfunc 1 2 3

echo "x is $x"

Script was called with

x is 1

I was called as : 1 2 3

the no of arguments passed : 3

1 2 3

x is 2