

Program - (1)

Check whether the given ~~no~~ ^{year} is leap year or not
and the given number is odd or even.

→ echo Enter the year

read year

if [`expr \$year % 4` -eq 0]

then

echo Leap year

else

echo Not a leap year

fi

→ echo Enter a number

read num

if [`expr \$num % 2` -eq 0]

then

echo The number is a even number

else

echo The number is odd

fi

Output - → Enter the year

2002

Not a leap year

→ Enter a number

2022

The number is a even number

Program - 04

Program to find the LCM and GCD of a number.

→ echo Enter the two numbers m and n

read m

read n

temp = `expr \$m * \$n'

echo the value of temp is \$ temp

while [\$m -ne \$n]

do

if [\$m -gt \$n]

then

m= `expr \$m - \$n'

else

n= `expr \$n - \$m'

fi

done

Output -

enter the two numbers m and n

5

6

the value of temp is 30

$$\text{lcm} = 30$$

$$\text{gcd} = 1$$

Program - (5)

Program for a use of break and continue

→ echo The program of using the break statement
using while and if conditions.

echo Enter a number

read Number

while [\$number -lt 10]

do

echo Value of number is \$number

if [\$number -eq 5]

then

break

fi

a = `expr \$number + 1'

done

Program - (6)

Program to print the string in the reverse order.

→ echo Enter the string and print it in the reverse order

read str

len = `echo \$ str | wc -c'

echo \$ len

len = `expr \$ len - 1'

echo Length of the string \$ len

echo Original string : \$ str

echo Length of string is : \$ len

echo Reversed string is

while [\$len -ge 1]

do

rev = `echo \$ str | cut -c \$ len'

Output - Enter the string and print it in
the reverse order

vonsh

length of the string 5

original string : vonsh

length of string is : 5

reversed string is

h

s

n

a

v

Program - 07

Program to calculate the electricity bill.

→ echo Enter the units consumed

echo Enter the units :

if [\$units -gt 10 -a \$units -lt 15]

then

echo units should be more

elif [\$units -gt 20]

then

net = `expr \$units * 2 + 25'

echo The value of units is \$ net

elif [\$units -gt 40]

then

net1 = `expr \$units * 5 + 25'

echo the value of units is \$ net1

else

echo units are crossed

fi

~~Program - 8~~

To check whether the string is a palindrome or not.

→ echo Enter a string

cat > str

Size = `cat str | wc -c'

len = `expr \$ size - 1'

len = `expr \$ len / 2'

i = 1

while [\${i} -le \$ len]

do

first = `cat str | cut -c \${i}'

ch = `expr \$ len - \${i} + 1'

last = `cat str | cut -c \${ch}'

if [\${first} \${i} = \${last}]

then

echo 'cat str' is not a palindrome

exit

fi

i = `expr \${i} + 1'

done

echo 'cat str' is a palindrome

Program - 9

Program to find the frequency of occurrence of a character.

→ echo Enter a string

cat > str

echo Enter the character to be searched

head chau

i = 1

count = 0

len = `cat str | wc -c'

echo The length of the string is \$ len

while [\${i} -lt \${len}]

do

ch = `cat str | cut -c \$'

if [\${ch} -eq \${chau}]

then

count = `expr \$ count + 1'

fi

i = `expr \${i} + 1'

done

echo The given string is `cat str'

echo \$ count times

Program - 10

Program to find the prime number in given ranges.

read m n

while [$\$m -lt \n]

do

$i = 2$

flag = 1

while [$\$i -lt \m]

do

if [$'expr \$m \% \$i' -eq 0$]

then

flag = 0

break

else

$i = 'expr \$i + 1'$

fi

done

if [$\$flag -eq 1$]

then

echo \$m

fi

$m = 'expr \$m + 1'$

done

Program - 11

Write the program to display the fibonacci series

→ echo Enter the limit
read n

a = 0

b = 1

echo the fibonacci series upto \$n is :

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

do

echo "\$a"

fib=\$((a+b))

a=\$b

b=\$fib

done

Output - The fibonacci series upto 5 is

0

1

1

2

3

5

Program - 12

Program for student marks, sum and average calculation.

→ echo Enter the marks of given five students
 ↗ head m₁
 ↗ head m₂
 ↗ head m₃
 ↗ head m₄
 ↗ head m₅

$$\text{Sum} = \text{expr } \$m_1 + \$m_2 + \$m_3 + \$m_4 + \$m_5$$

echo The sum of the marks is \$ sum

$$\text{avg} = \text{expr } \$\text{sum} / 5$$

echo The value of average is \$ avg

if [\$m₁ -lt 40 -o \$m₂ -lt 40 -o \$m₃
 - lt 40 -o \$m₄ - lt 40]

then

echo fail

elif [\$avg -ge 70]

then

echo [Distinction]

elif [\$avg -ge 60]

then

echo First Class

elif [\$avg -ge 50]

then

echo Second Class

elif [\$avg -ge 40]

then

echo Third class

else

echo Not Pressed # fi

fi

Program - 13

Programs for Arithmetic operations using 'expr'
and 'bc' .

→ echo Enter the value of a and b

head a

head b

c = `echo \$a + \$b | bc'

d = `echo \$a - \$b | bc'

e = `echo \$a * \$b | bc'

f = `echo \$a / \$b | bc'

g = `echo \$a % \$b | bc'

echo Addition of two no. is \$c

echo Subtraction of two no. is \$d

echo Multiplication of two no. is \$e

echo Division of two no. is \$f

echo Modulus of two no. is \$g

echo The program to use the usage of basic
calculation

head c

head d

scale = 2

res = `echo \$c + \$d | bc'

res1 = `echo \$c - \$d | bc'

res2 = `echo \$c / \$d | bc'

res3 = `echo \$c % \$d | bc'

echo a + b = \$res

echo a - b = \$res1

echo a/b = \$ mes 4

echo a%b = \$ mes 5

Program - 15

Program to find vowels of a given string.

→ echo Enter a string

cat > str

echo The entered string is \$ str

len = cat str | wc -c

echo length of the string is \$ len

i = 1

actu = 0 ; ectu = 0 ; ictu = 0 ; octu = 0 ; uctu = 0 ; other =

while [\$i -le \$len]

do

ch = `cat str | cut -c \${i}'

case \$ch in

a | A) actu = `expr \$actu + 1'

e | E) ectu = `expr \$ectu + 1'

i | I) ictu = `expr \$ictu + 1'

o | O) octu = `expr \$octu + 1'

u | U) uctu = `expr \$uctu + 1'

*) other = `expr \$other + 1'

esac

i = `expr \${i} + 1'

done

echo Number of a or A = \$actu

echo Number of e or E = \$ectu

echo Number of i or I = \$ictu

echo Number of o or O = \$octu

echo Number of u or U = \$uctu

echo Number of other character = \$other

Program - 17

Program to find the sum of given digit in a given number.

→ Sum = 0

echo " enter a no "

read num

n = \$ sum

while [\$ num -gt 0]

do

rem = `expr \$ num % 10'

sum = `expr \$ sum + \$ rem'

num = `expr \$ num / 10 '

done

echo The sum of digits is \$ sum

Program - 18

Program to list the files in the current directory.

→ for file in *

do

if [-r \$file -a -w \$file]

then

echo "\$file file is readable and writeable"

elif [-w \$file]

then

echo "\$file file is only writeable"

elif [-r \$file]

then

echo "\$file file is only readable"

elif [-d \$file]

then

echo "\$file is a directory"

fi

done

Program - (19)

Program to create a file and compress it.

→ echo " Enter the file name to be compressed

read fname

wc - c \$ fname

gzip \$ fname

gzip -1 \$ fname.gz

echo " Decompressing the file "

gunzip \$ fname.gz

wc - c \$ fname

Program - 20

Write a program to convert the contents of given text from uppercase to lowercase and count the number of characters, lines, words of the resultant text.

→ fn = "bcde"

for i in \$fn

do

i='echo \$i | tr '[a-z]' '[A-Z]'

echo \$i

(

done

2) Programs to find the largest of 3 numbers.

echo "Enter 3 numbers"

read n1

read n2

read n3

if [\$n1 -gt \$n2]

then

if [\$n1 -gt \$n3]

then

echo n1 is greater than both

else

fi

else

if [\$n2 -gt \$n3]

then

echo n2 is greater

else

echo n3 is greater

fi

fi

2) Data - Output

Enter 3 numbers

1

2

3

n_3 is greater

3) Program to find the multiplication table using while loops and to find the sum of natural numbers.

echo Read a number

read n

echo Enter the value of i

read i

while [\$i -le 10]

do

echo for $i^* \text{$i}$ is : 'expr $\$i * \i '

$i = 'expr \$i + 1'$

done

echo Read the value of n

read n

echo Enter the value of i

read i

while [\$i -le \$n]

do

echo The value of i is $\$i$

$i = 'expr \$i + 1'$

done

3) Output

Read a number

2

Enter the value of i

8

$2 * 8 \text{ is : } 16$

$2 * 9 \text{ is : } 18$

$2 * 10 \text{ is : } 20$

Enter read the value of n

4

Enter the value of i

2

The value of i is 2

The value of i is 3

The value of i is 4