1. Use Random function ((RANDOM )) to get single digit

Shell code:

#!/bin/bash -x

a=$((RANDOM%10))

echo $a

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./singledigit.sh

+ a=2

+ echo 2

2

1. Use random to get dice number between 1 to 6

Shell code:

#!/bin/bash -x

a=$((RANDOM%6))

if [ $a -eq 0 ]

then

echo "6"

else

echo $a

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./inbetwin1to6.sh

+ a=2

+ '[' 2 -eq 0 ']'

+ echo 2

2

1. Add two random dice number and print the result

Shell code:

#!/bin/bash -x

a=$((RANDOM%6))

if [ $a -eq 0 ]

then

a=6

echo "Dice first value : " $a

else

echo "Dice first value : " $a

fi

b=$((RANDOM%6))

if [ $b -eq 0 ]

then

b=6

echo "Dice second value : " $b

else

echo "Dice second value : " $b

fi

echo " Addition of dice number" `expr $a + $b`

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./adddicenumber.sh

+ a=5

+ '[' 5 -eq 0 ']'

+ echo 'Dice first value : ' 5

Dice first value : 5

+ b=2

+ '[' 2 -eq 0 ']'

+ echo 'Dice second value : ' 2

Dice second value : 2

++ expr 5 + 2

+ echo ' Addition of dice number' 7

Addition of dice number 7

1. Write a program that read 5 random 2 digit values , then find their sum and the average

Shell code:

##!/bin/bash -x

echo "Enter the Total no you want to enter:"

n=5

i=0

while [ $i -lt $n ]

do

a[$i]=$((RANDOM%100))

i=`expr $i + 1`

done

sub=0

echo "Output :"

for i in "${a[@]}"

do

echo $i

sub=`expr $sub + $i`

#echo "Addition of five number : " $sub

done

add=o

echo "Addition of five number : " $sub

avg=`expr $sub / $n`

echo "Average of five number : " $avg

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./fivenumaddandavg.sh

Enter the Total no you want to enter:

Output :

80

8

9

43

33

Addition of five number : 173

Average of five number : 34

1. Unit conversion
   1. 1ft=12in then 42in=?ft

Shell code:

#!/bin/bash -x

echo "Enter 1 for convesion in feet or enter 0"

read a

if [ $a -eq 1 ]

then

echo "Enter value : "

read b

c=`scale=5 expr $b / 12`

echo "$a inch = $c feet"

else

echo "Enter the value : "

read d

f=12

e=`expr $d \\* $f`

echo "$d feet = $e inch"

"

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./fttoin.sh

+ echo 'Enter 1 for convesion in feet or enter 0'

Enter 1 for convesion in feet or enter 0

+ read a

1

+ '[' 1 -eq 1 ']'

+ echo 'Enter value : '

Enter value :

+ read b

42

++ scale=5

++ expr 42 / 12

+ c=3

+ echo '1 inch = 3 feet'

1 inch = 3 feet

* 1. Rectangular plot of 60 feet X 40 feet in meters

Shell code:

#!/bin/bash -x

echo "Read the lenth of rectangular part feet : "

read l

echo "Read the bredth of rectangular part feet: "

read b

a=$(($l \* $b))

echo "Area of plot in feet $a"

meterCon=0.092903

meter=$(awk 'BEGIN {print '$a' \* '$meterCon'}')

echo " Area in meters $meter"

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./feetintometer.sh

+ echo 'Read the lenth of rectangular part feet : '

Read the lenth of rectangular part feet :

+ read l

40

+ echo 'Read the bredth of rectangular part feet: '

Read the bredth of rectangular part feet:

+ read b

60

+ a=2400

+ echo 'Area of plot in feet 2400'

Area of plot in feet 2400

+ meterCon=0.092903

++ awk 'BEGIN {print 2400 \* 0.092903}'

+ meter=222.967

+ echo ' Area in meters 222.967'

Area in meters 222.967

* 1. Calculate area of 25 such plots in acres

Shell code:

##!/bin/bash -x

echo "Read the lenth of rectangular part feet : "

read l

echo "Read the bredth of rectangular part feet: "

read b

a=$(($l \* $b))

echo "Area of plot in feet $a"

meterCon=0.092903

meter=$(awk 'BEGIN {print '$a' \* '$meterCon' \* '25'}')

echo "Area of 25 plots in meters $meter"

Sqmetertoacre=0.000247105

acre=$(awk 'BEGIN {print '$meter' \* '$Sqmetertoacre'}')

echo "In acres $acre"

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./acres.sh

Read the lenth of rectangular part feet :

40

Read the bredth of rectangular part feet:

60

Area of plot in feet 2400

Area of 25 plots in meters 5574.18

In acres 1.37741

1. Write a program that reads 5 random 3 digit values and then output the minimum and the maximum value

Shell code:

##!/bin/bash -x

echo "Enter the Total no you want to enter:"

n=5

i=0

while [ $i -lt $n ]

do

a[$i]=$((RANDOM%1000))

i=`expr $i + 1`

done

echo "Output :"

for i in "${a[@]}"

do

echo $i

done

max=0

min=10000

for i in "${a[@]}"

do

if [ $i -gt $max ]

then

max=$i

fi

if [ $i -lt $min ]

then

min=$i

fi

done

echo "Maximum number is : " $max

echo "Minimum number is : " $min

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./minmaxrandom.sh

Enter the Total no you want to enter:

Output :

194

436

105

832

168

Maximum number is : 832

Minimum number is : 105

1. Write a program that takes day and month from the command line and prints true if day of month is between March 20 and june 20, false otherwise.

Shell code:

#!/bin/bash -x

date1="0319"

date2="0621"

echo "please enter the date in mmdd formate:"

read x

if [ $x -gt $date1 -a $x -lt $date2 ]

then

echo "True"

else

echo "False"

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./checkdate.sh

+ date1=0319

+ date2=0621

+ echo 'please enter the date in mmdd formate:'

please enter the date in mmdd formate:

+ read x

0620

+ '[' 0620 -gt 0319 -a 0620 -lt 0621 ']'

+ echo True

True

1. Write a program that takes a year as input and output the year is a leap year or not a leap year. A leap year checks from 4 digit number, divisible by 4 and not 100 unless divisible by 400.

Shell code:

#!/bin/bash -x

echo "Enter the year you want to check :"

read year

a=`expr $year % 400`

b=`expr $year % 4`

c=`expr $year % 100`

zero=0

if (($b==$zero))

then

if (($a==$zero))

then

if (($c==$zero))

then

echo "$year is leap year."

fi

fi

else

echo "$year is not leap year."

fi

Output :

$ ./checkleapyear.sh

+ echo 'Enter the year you want to check :'

Enter the year you want to check :

+ read year

2000

++ expr 2000 % 400

+ a=0

++ expr 2000 % 4

+ b=0

++ expr 2000 % 100

+ c=0

+ zero=0

+ (( 0==0 ))

+ (( 0==0 ))

+ (( 0==0 ))

+ echo '2000 is leap year.'

2000 is leap year.

1. Write a program to simulate a coin flip and print out “Heads” or “Tails” accordingly.

Shell code:

#!/bin/bash -x

a=$((RANDOM%10))

echo $a

Q=`expr $a % 2`

if [ $Q -eq 0 ]

then

echo "Head"

else

echo "Tail"

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./haedstails.sh

+ a=1

+ echo 1

1

++ expr 1 % 2

+ Q=1

+ '[' 1 -eq 0 ']'

+ echo Tail

Tail

1. Read a single digit number and write the number in word

Shell code:

#!/bin/bash -x

echo "Enter the single digit number : "

read n

if [ $n -eq 0 ]

then

echo "Zero"

elif [ $n -eq 1 ]

then

echo "One"

elif [ $n -eq 2 ]

then

echo "Two"

elif [ $n -eq 3 ]

then

echo "Three"

elif [ $n -eq 4 ]

then

echo "Four"

elif [ $n -eq 5 ]

then

echo "Five"

elif [ $n -eq 6 ]

then

echo "Six"

elif [ $n -eq 7 ]

then

echo "Seven"

elif [ $n -eq 8 ]

then

echo "Eight"

elif [ $n -eq 9 ]

then

echo "Nine"

else

echo "This is not single digit number."

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./singletoword.sh

+ echo 'Enter the single digit number : '

Enter the single digit number :

+ read n

7

+ '[' 7 -eq 0 ']'

+ '[' 7 -eq 1 ']'

+ '[' 7 -eq 2 ']'

+ '[' 7 -eq 3 ']'

+ '[' 7 -eq 4 ']'

+ '[' 7 -eq 5 ']'

+ '[' 7 -eq 6 ']'

+ '[' 7 -eq 7 ']'

+ echo Seven

Seven

1. Read a Number and Display the week day (Sunday, Monday,...)

Shell code:

#!/bin/bash -x

echo "Enter the single digit number : "

read n

if [ $n -eq 1 ]

then

echo "Sunday"

elif [ $n -eq 2 ]

then

echo "Monday"

elif [ $n -eq 3 ]

then

echo "Tuseday"

elif [ $n -eq 4 ]

then

echo "Wednesday"

elif [ $n -eq 5 ]

then

echo "Thursday"

elif [ $n -eq 6 ]

then

echo "Friday"

elif [ $n -eq 7 ]

then

echo "Saturday"

else

echo "This is not week day."

fi

Output :

$ ./weekdayif.sh

+ echo 'Enter the single digit number : '

Enter the single digit number :

+ read n

4

+ '[' 4 -eq 1 ']'

+ '[' 4 -eq 2 ']'

+ '[' 4 -eq 3 ']'

+ '[' 4 -eq 4 ']'

+ echo Wednesday

Wednesday

1. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,...

Shell code:

#!/bin/bash -x

echo "Enter the single digit number : "

read n

if [ $n -eq 1 ]

then

echo "Unit"

elif [ $n -eq 10 ]

then

echo "Ten"

elif [ $n -eq 100 ]

then

echo "Hundred"

elif [ $n -eq 1000 ]

then

echo "Thousand"

elif [ $n -eq 10000 ]

then

echo "Ten Thousand"

elif [ $n -eq 100000 ]

then

echo "Hundred Thousand"

elif [ $n -eq 1000000 ]

then

echo "Million"

elif [ $n -eq 10000000 ]

then

echo "Ten Million"

elif [ $n -eq 100000000 ]

then

echo "Hundred Million"

elif [ $n -eq 1000000000 ]

then

echo "Billion"

else

echo "Greter than billion."

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./disten.sh

+ echo 'Enter the single digit number : '

Enter the single digit number :

+ read n

100000

+ '[' 100000 -eq 1 ']'

+ '[' 100000 -eq 10 ']'

+ '[' 100000 -eq 100 ']'

+ '[' 100000 -eq 1000 ']'

+ '[' 100000 -eq 10000 ']'

+ '[' 100000 -eq 100000 ']'

+ echo 'Hundred Thousand'

Hundred Thousand

1. Enter 3 Numbers do following arithmetic operation and find the one that

is maximum and minimum

* 1. a + b \* c
  2. a % b + c
  3. c + a / b
  4. a \* b + c

Shell code:

echo "Enter Three numbers : "

read a

read b

read c

arry[0]=$(awk 'BEGIN {print '$a' + '$b' \* '$c'}')

arry[1]=$(awk 'BEGIN {print '$a' % '$b' + '$c'}')

arry[2]=`expr $a + $b / $c`

a=`expr $a + $b / $c`

echo $a

arry[3]=$(awk 'BEGIN {print '$a' \* '$b' + '$c'}')

echo "1. a + b \* c = ${arry[0]}"

echo "2. a % b + c = ${arry[1]}"

echo "3. c + a / b = ${arry[2]}"

echo "4. a \* b + c = ${arry[3]}"

max=0

min=10000

for i in "${arry[@]}"

do

if [ $i -gt $max ]

then

max=$i

fi

if [ $i -lt $min ]

then

min=$i

fi

done

echo "Maximum value in above arethmatic opration : $max"

echo "Minimum value in above arethmatic opration : $min"

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./arithmatic.sh

Enter Three numbers :

1

5

8

1

1. a + b \* c = 41

2. a % b + c = 9

3. c + a / b = 1

4. a \* b + c = 13

Maximum value in above arethmatic opration : 41

Minimum value in above arethmatic opration : 1

1. Read a single digit number and write the number in word using Case

Shell code:

#!/bin/bash -x

echo "Enter the single digit number : "

read n

case "$n" in

0) echo "Zero"

;;

1) echo "One"

;;

2) echo "Two"

;;

3) echo "Three"

;;

4) echo "Four"

;;

5) echo "Five"

;;

6) echo "Six"

;;

7) echo "Seven"

;;

8) echo "Eight"

;;

9) echo "Nine"

;;

\*) echo "This is not single digit number."

;;

esac

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./digittowordcase.sh

+ echo 'Enter the single digit number : '

Enter the single digit number :

+ read n

5

+ case "$n" in

+ echo Five

Five

1. Read a Number and Display the week day (Sunday, Monday,...)

Shell code:

#!/bin/bash -x

echo "Enter the single digit number : "

read n

case "$n" in

1) echo "Sunday"

;;

2) echo "Monday"

;;

3) echo "Tuseday"

;;

4) echo "Wednesday"

;;

5) echo "Thursday"

;;

6) echo "Friday"

;;

7) echo "Saturday"

;;

\*) echo "This is not week day."

;;

esac

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./weekdaycase.sh

+ echo 'Enter the single digit number : '

Enter the single digit number :

+ read n

6

+ case "$n" in

+ echo Friday

Friday

1. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,...

Shell code:

#!/bin/bash -x

echo "Enter the single digit number : "

read n

case "$n" in

1) echo "Unit"

;;

10) echo "Ten"

;;

100) echo "Hundred"

;;

1000) echo "Thousand"

;;

10000) echo "Ten Thousand"

;;

100000) echo "Hundred Thousand"

;;

1000000) echo "Million"

;;

10000000) echo "Ten Million"

;;

100000000) echo "Hundred Million"

;;

1000000000) echo "Billion"

;;

\*) echo "Greter than billion."

;;

esac

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./unitcase.sh

+ echo 'Enter the single digit number : '

Enter the single digit number :

+ read n

10000

+ case "$n" in

+ echo 'Ten Thousand'

Ten Thousand

1. Write a program that takes User Inputs and does Unit Conversion of

different Length units

* 1. Feet to Inch
  2. Feet to Meter
  3. Inch to Feet
  4. Meter to Feet

Shell code:

##!/bin/bash -x

echo "1. Feet to Inch"

echo "2. Feet to Meter"

echo "3. Inch to Feet"

echo "4. Meter to Feet"

read n

case "$n" in

1)

echo "Enter the value in feet : "

read a

inch=$(awk 'BEGIN {print '$a' \* '12'}')

echo "$a feet= $inch inch"

;;

2)

echo "Enter the value in feet : "

read a

inch=$(awk 'BEGIN {print '$a' \* '0.3048'}')

echo "$a feet= $inch meter"

;;

3)

echo "Enter the value in Inch : "

read a

inch=$(awk 'BEGIN {print '$a' \* '0.0833333'}')

echo "$a inch= $inch feet"

;;

4)

echo "Enter the value in Meter : "

read a

inch=$(awk 'BEGIN {print '$a' \* '3.28084'}')

echo "$a meter= $inch feet"

;;

\*) echo "Invalid case."

;;

esac

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile

$ ./unitconvercase.sh

1. Feet to Inch

2. Feet to Meter

3. Inch to Feet

4. Meter to Feet

4

Enter the value in Meter :

40

40 meter= 131.234 feet