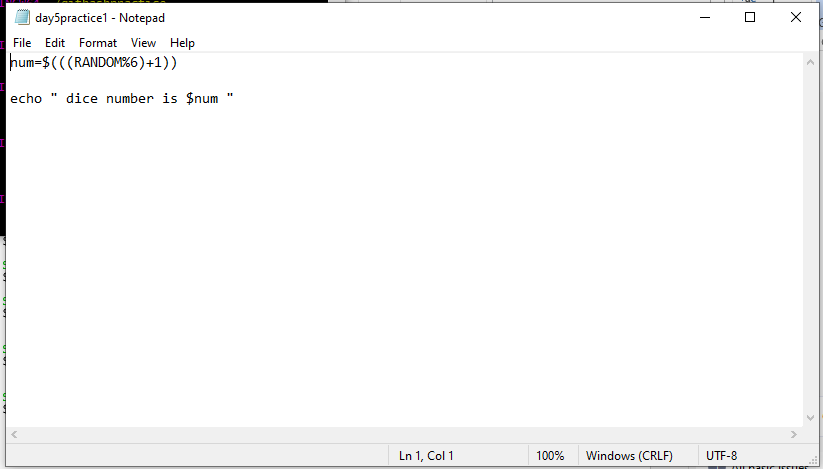
**DAY 5 PRACTICE PROBLEM**

**1. Use Random to get Dice Number between 1 to 6**

1. STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice
2. $ touch day5practice1.sh
3. STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice
4. $ notepad day5practice1.sh
5. STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice
6. $ bash day5practice1.sh
7. dice number is 4
8. STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice
9. $ bash day5practice1.sh
10. dice number is 1
11. STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice
12. $

****

**2.** **Use Random Function (( RANDOM )) to get Single Digit**

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ touch day5practice2.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad day5practice2.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5practice2.sh

RANDOM Number is 31302

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad day5practice2.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

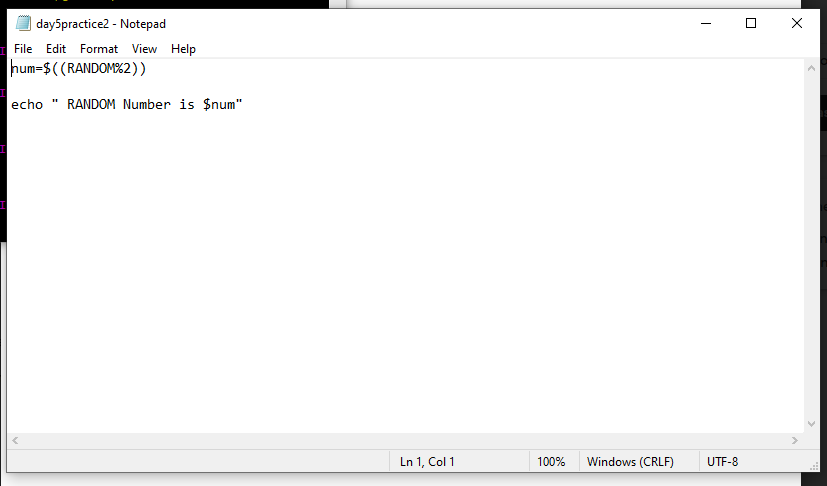
$ bash day5practice2.sh

RANDOM Number is 0

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5practice2.sh

RANDOM Number is 0

****

**3.** **Add two Random Dice Number and Print the Result**

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad day5practice2.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5practice2.sh

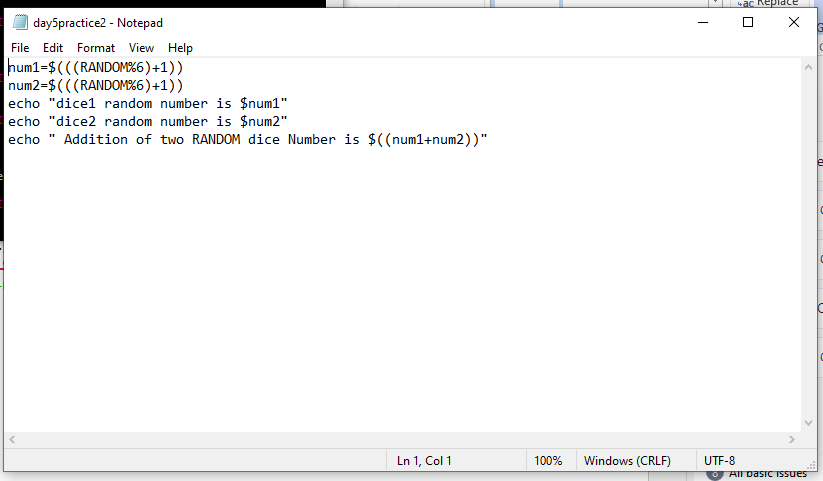
dice1 random number is 1

dice2 random number is 2

Addition of two RANDOM dice Number is 3

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$

****

**4.Write a program that reads 5 Random 2 Digit values , then find their**

**sum and the average**

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ touch day5practice4que.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad day5practice4que.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5practice4que.sh

random number1 is 46

random number2 is 45

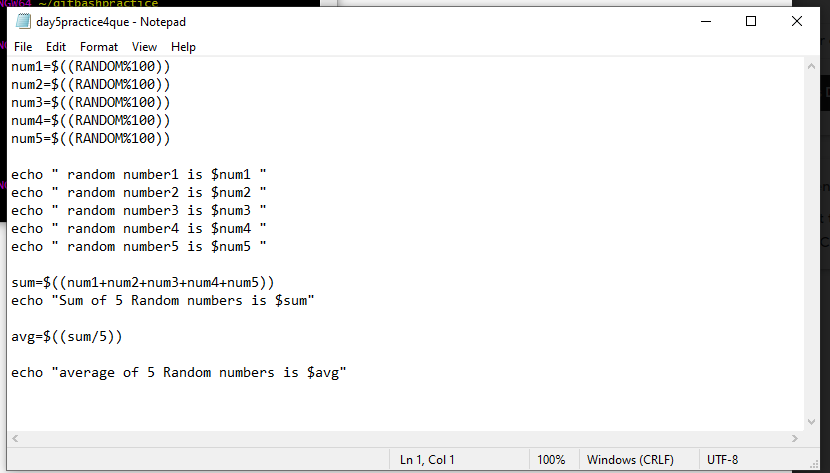
random number3 is 96

random number4 is 3

random number5 is 18

Sum of 5 Random numbers is 208

average of 5 Random numbers is 41

****

**5. Unit Conversion**

**a. 42 inch= ?feet**

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ touch day5practice5que.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

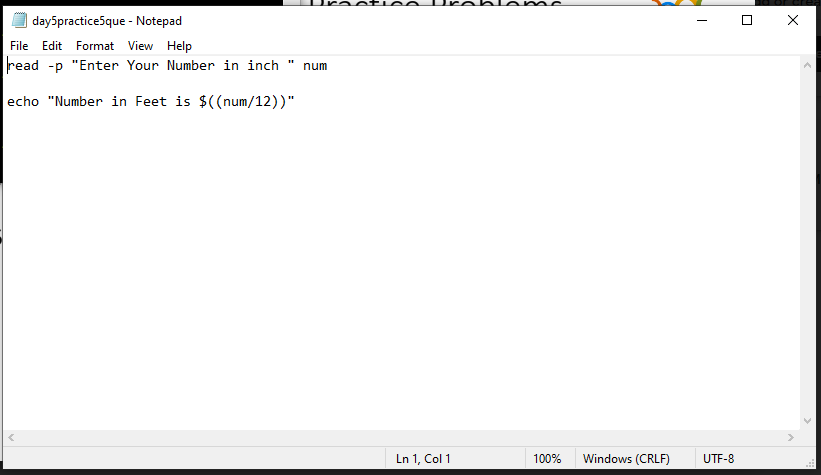
$ notepad day5practice5que.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5practice5que.sh

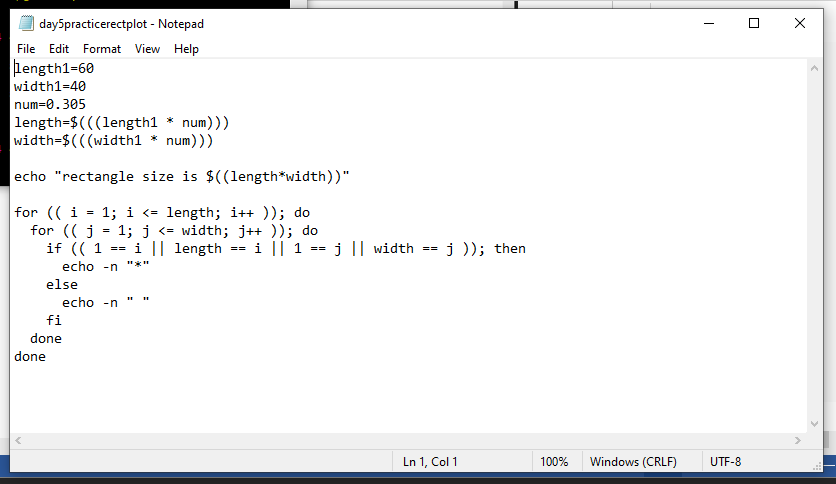
Enter Your Number in inch 42

Number in Feet is 3

****

**b. Rectangular Plot of 60 feet x 40 feet in meters**

**not work**

****

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ touch rectangle.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad rectangle.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash rectangle.sh

3

Area of rectangle:2400

**6.Write program that reads 5 Random 3 Digit values and then outputs the minimum and the maximum value**

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ touch day5pq1.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad day5pq1.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad day5pq1.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5pq1.sh

146

703

571

856

811

max number is :856

min number is :0

**7.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.**

read -p "Enter Date: " date

read -p "Enter Month: " Month

combo=0

res="false"

st=1

if [ $Month -ge 3 -a $Month -le 6 ]

then

dlimit=$((30+(Month%2)))

if [ $date -ge 1 -a $date -le $dlimit ]

then

combo=$(((Month\*100)+date))

if [ $combo -ge 320 -a $combo -le 620 ]

then

res="true"

st=0

fi

fi

fi

echo "$Month/$date $res"

exit $st

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ touch day5pq2.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ notepad day5pq2.sh

STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5pq2.sh

Enter Date: 30

Enter Month: 2

2/30 false

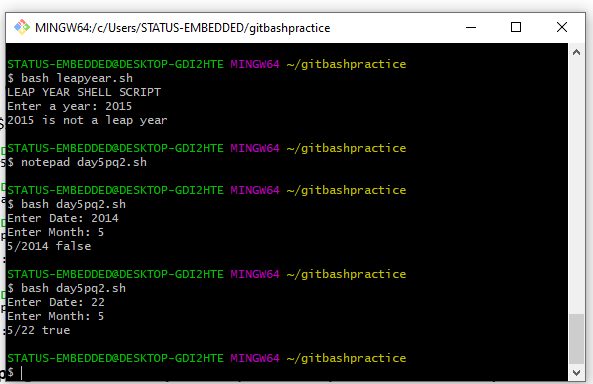
STATUS-EMBEDDED@DESKTOP-GDI2HTE MINGW64 ~/gitbashpractice

$ bash day5pq2.sh

Enter Date: 30

Enter Month: 4

4/30 true



**8.Write a program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year. A Leap Year checks for 4 Digit Number, Divisible by 4 and not 100 unless divisible by 400.**

Answer:

echo "LEAP YEAR SHELL SCRIPT"

echo -n "Enter a year: "

read year\_checker

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

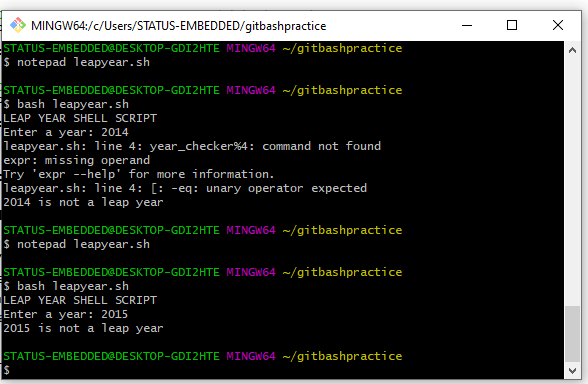
then

echo "$year\_checker is a leap year"

else

echo "$year\_checker is not a leap year"

fi

****

**Write a program to simulate a coin flip and print out "Heads" or "Tails" accordingly.**

**Answer:**

Result=$((RANDOM%2))

if [ $Result == 0 ]

then

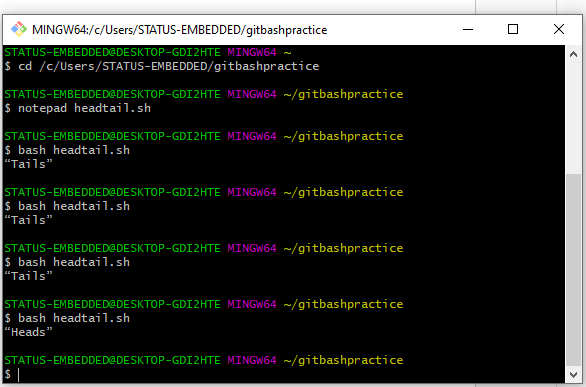
echo “Heads”

elif [ $Result == 1 ]

then

echo “Tails”

fi



IF …Else Problem of day 5

1. **Read a single digit number and write the number in word Using If …else**

**Answer:**

read -p "Enter a number between 0 and 9 inclusive > " number

if [ "$number" = "1" ]; then

echo "One"

elif [ "$number" = "2" ]; then

echo "Two"

elif [ "$number" = "3" ]; then

echo "Three"

elif [ "$number" = "4" ]; then

echo "Four"

elif [ "$number" = "5" ]; then

echo "Five"

elif [ "$number" = "6" ]; then

echo "Six"

elif [ "$number" = "7" ]; then

echo "Seven"

elif [ "$number" = "8" ]; then

echo "Eight"

elif [ "$number" = "9" ]; then

echo "Nine"

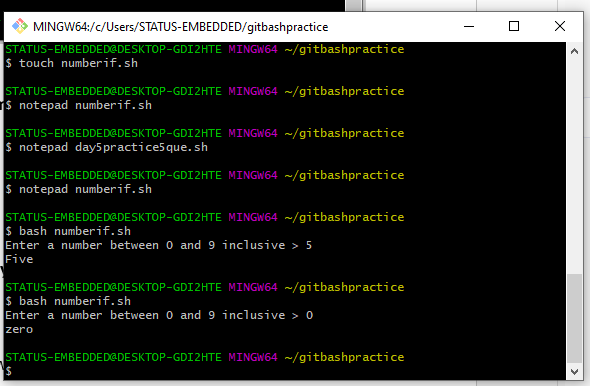
elif [ "$number" = "0" ]; then

echo "zero"

else

echo "You did not enter a number between 1 and 7."

fi

****

**2.Read a Number and Display the week day (Sunday, Monday,...) Using IF…….Else**

**Answer:**

read -p "Enter a number between 1 and 7 inclusive > " number

if [ "$number" = "1" ]; then

echo "Sunday"

elif [ "$number" = "2" ]; then

echo "Monday"

elif [ "$number" = "3" ]; then

echo "tuesday"

elif [ "$number" = "4" ]; then

echo "wednesday"

elif [ "$number" = "5" ]; then

echo "Thirsday"

elif [ "$number" = "6" ]; then

echo "Friday"

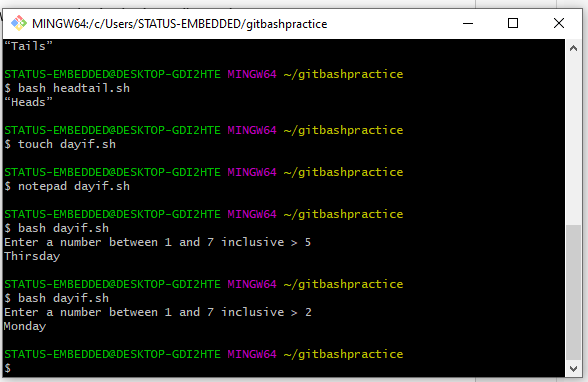
elif [ "$number" = "7" ]; then

echo "saturday"

else

echo "You did not enter a number between 1 and 7."

Fi

**3.Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,... Using if…else**

**Answer:**

read -p "Enter a number in multiple of ten start from 1 > " number

if [ "$number" = "1" ]; then

echo "Unit"

elif [ "$number" = "10" ]; then

echo "Ten"

elif [ "$number" = "100" ]; then

echo "Hundred"

elif [ "$number" = "1000" ]; then

echo "Thousand"

elif [ "$number" = "10000" ]; then

echo "Ten Thousand"

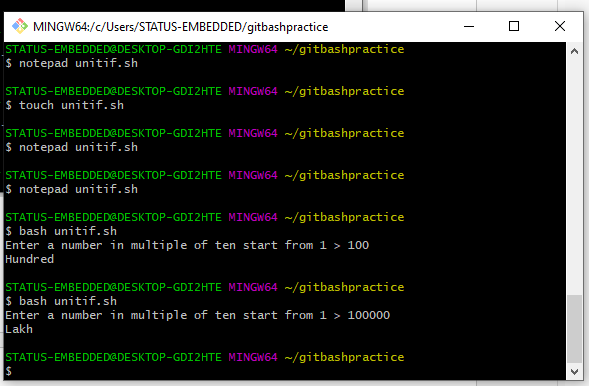
elif [ "$number" = "100000" ]; then

echo "Lakh"

else

echo "You did not enter a number in multiple of ten"

fi



**4.** **Enter 3 Numbers do following arithmetic operation and find the one that is maximum and minimum**

**1. a + b \* c**

echo "Enter Num1"

read num1

echo "Enter Num2"

read num2

echo "Enter Num3"

read num3

result=$(((num1+num2)\*num3))

echo "a+b\*c =$result"

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]

then

echo $num1

elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]

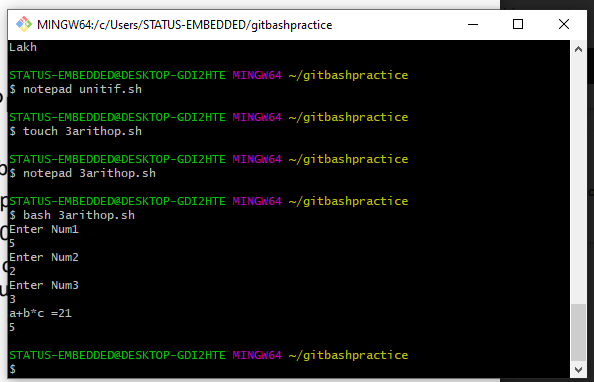
then

echo $num2

else

echo $num3

fi



**Enter 3 Numbers do following arithmetic operation and find the one that is maximum and minimum**

**1. a + b \* c 3. c + a / b**

**2. a % b + c 4. a \* b + c**

echo "Enter Num1"

read num1

echo "Enter Num2"

read num2

echo "Enter Num3"

read num3

result1=$(((num1+num2)\*num3))

result2=$(((num1%num2)+num3))

result3=$((num3+(num1/num2)))

result4=$(((num1\*num2)+num3))

echo "a+b\*c =$result1"

echo "a%b+c=$result2"

echo "c+a/b=$result3"

echo "a\*b+c=$result4"

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]

then

echo $num1

elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]

then

echo $num2

else

echo $num3

fi

if [ $num1 -lt $num2 ] && [ $num1 -lt $num3 ]

then

echo $num1

elif [ $num2 -lt $num1 ] && [ $num2 -lt $num3 ]

then

echo $num2

else

echo $num3

fi

**With case statement:**

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

read -p "Enter a number between 1 and 3 inclusive > " number

case $number in

1 ) echo "You entered one."

;;

2 ) echo "You entered two."

;;

3 ) echo "You entered three."

;;

4 ) echo "You entered Four."

;;

5 ) echo "You entered Five."

;;

6 ) echo "You entered Six."

;;

7 ) echo "You entered Seven."

;;

8 ) echo "You entered Eight."

;;

9 ) echo "You entered Nine."

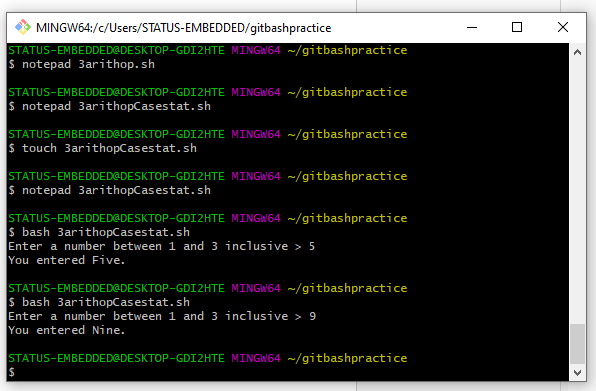
;;

0 ) echo "You entered Zero."

;;

\* ) echo "You did not enter a number between 0 and 9."

Esac



**Read a Number and Display the week day (Sunday, Monday,...) using Case Statement**

**Answer:**

read -p "Enter a number between 1 and 7 inclusive > " number

case $number in

1 ) echo "Sunday"

;;

2 ) echo "Monday"

;;

3 ) echo "Tuesday"

;;

4 ) echo "Wednesday"

;;

5 ) echo "thursday"

;;

6 ) echo "Friday"

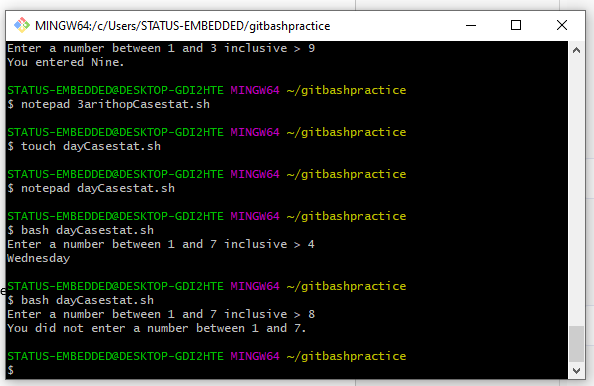
;;

7 ) echo "Saturday"

;;

\* ) echo "You did not enter a number between 1 and 7."

Esac



**Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,... using case statement**

**Answer:**

read -p "Enter a number in multiple of ten start from 1 > " number

case $number in

1 ) echo "Unit"

;;

10 ) echo "Ten"

;;

100 ) echo "Hundred"

;;

1000 ) echo "Thousand"

;;

10000 ) echo "Ten Thousand"

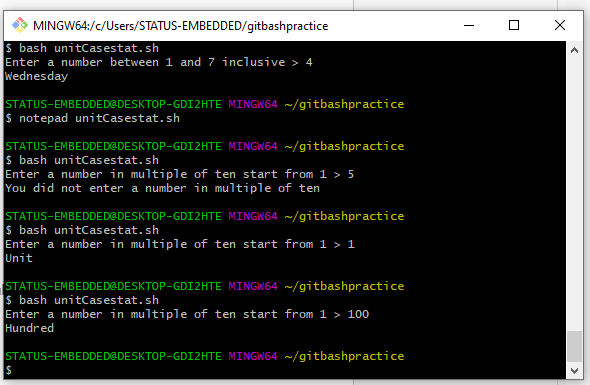
;;

100000 ) echo "Lakh"

;;

\* ) echo "You did not enter a number in multiple of ten"

Esac



**Write a program that takes User Inputs and does Unit Conversion of different Length units**

**1. Feet to Inch 3. Inch to Feet**

**2. Feet to Meter 4. Meter to Feet**

**Answer:**

read -p "Enter conversion type " src\_unit

read -p "Enter value " value

case $src\_unit in

feet-inch ) result=$(value \* 12)

echo "$result"

;;

inch-feet ) result=$(value / 12)

echo "$result"

;;

feet-meter ) result=$(value \* 0.305)

echo "$result"

;;

meter-feet ) result=$(value / 0.305)

echo "$result"

;;

\* ) echo "entered type conversion is not valid "

esac