

Day 5 problem 1 solution

1. Use Random Function ((RANDOM)) to get Single Digit

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ echo $RANDOM
16678
```

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

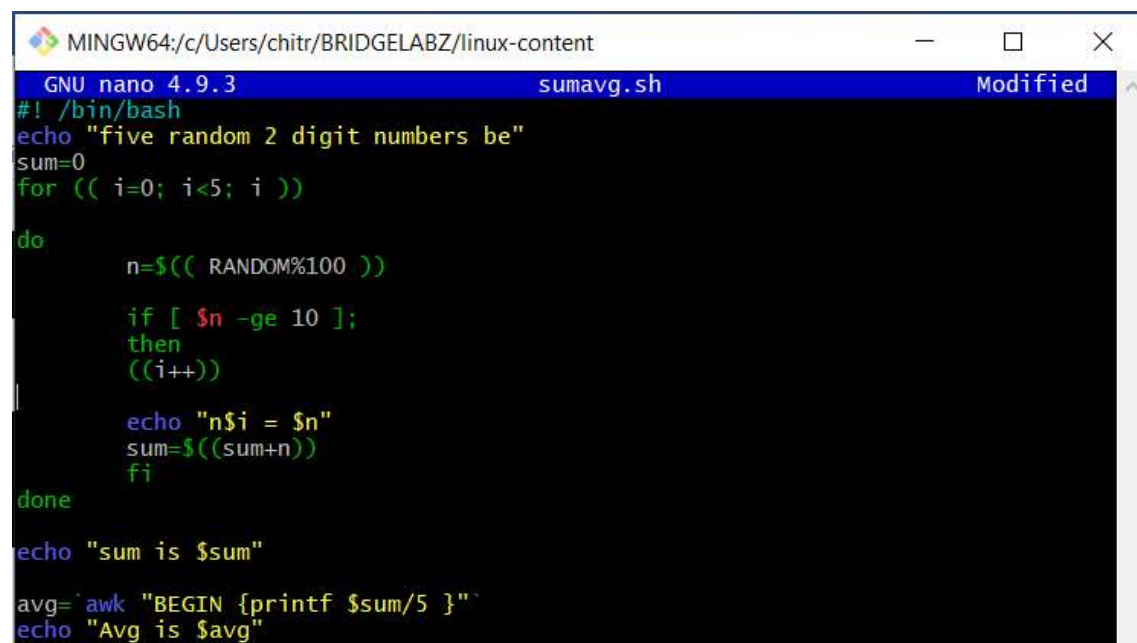
```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ echo $((RANDOM%6 + 1))
3
```

3. Add two Random Dice Number and Print the Result

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ echo $((RANDOM%6 + 1 + RANDOM%6 + 1))
9
```

4. Write a program that reads 5 Random 2 Digit values, then find their sum and the average

PROGRAM:



```
MINGW64:/c/Users/chitr/BRIDGELABZ/linux-content
GNU nano 4.9.3 sumavg.sh Modified
#!/bin/bash
echo "five random 2 digit numbers be"
sum=0
for (( i=0; i<5; i ))
do
    n=$(( RANDOM%100 ))
    if [ $n -ge 10 ];
    then
        ((i++))
    fi
    echo "n$i = $n"
    sum=$((sum+n))
done
echo "sum is $sum"
avg=`awk "BEGIN {printf $sum/5 }"`
echo "Avg is $avg"
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./sumavg.sh
five random 2 digit numbers be
n1 = 14
n2 = 41
n3 = 84
n4 = 91
n5 = 32
sum is 262
Avg is 52.4
```

5. Unit Conversion

- a. 1ft = 12 in then 42 in = ? ft
- b. Rectangular Plot of 60 feet x 40 feet in meters
- c. Calculate area of 25 such plots in acres

PROGRAM:

```
MINGW64:/c/Users/chitr/BRIDGELABZ/linux-content
GNU nano 4.9.3 unitconversions.sh
#!/bin/bash
echo "enter a, b or c"
read w

if [ $w = a ];
then
p=`awk "BEGIN {print 42/12}"`
echo -n "42 inch is $p feet"

elif [ $w = b ];
then
x=`awk "BEGIN {print 60*12.54/100}"`
y=`awk "BEGIN {print 40*12.54/100}"`
echo "60 feet x 40 feet is $x meter X $y meter"

elif [ $w = c ];
then
area=`awk "BEGIN {print 60*40*25*2.296/100000}"`
echo "Area of 25 plots of 60 feet X 40 feet is $area acre"

else
echo "letter you entered does not match"
fi
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./unitconversions.sh
enter a, b or c
m
letter you entered does not match

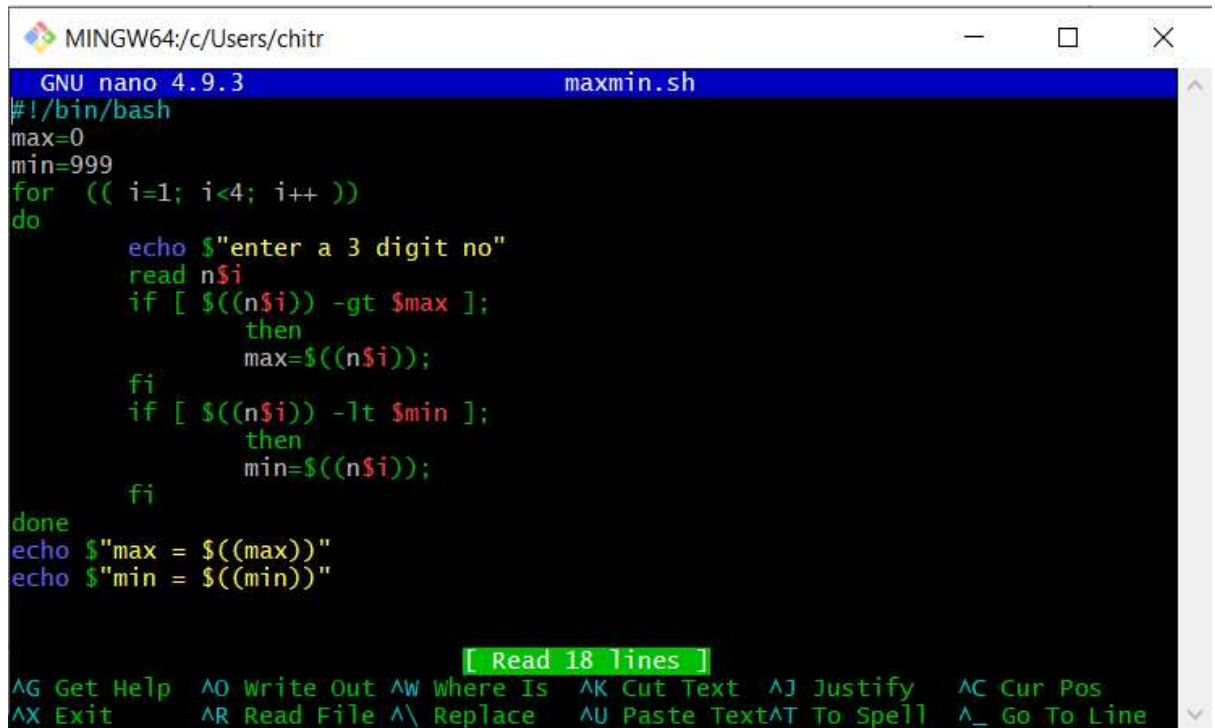
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./unitconversions.sh
enter a, b or c
c
Area of 25 plots of 60 feet X 40 feet is 1.3776 acre

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./unitconversions.sh
enter a, b or c
a
42 inch is 3.5 feet
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./unitconversions.sh
enter a, b or c
b
60 feet x 40 feet is 7.524 meter X 5.016 meter
```

Day 5 problem 2 solution

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

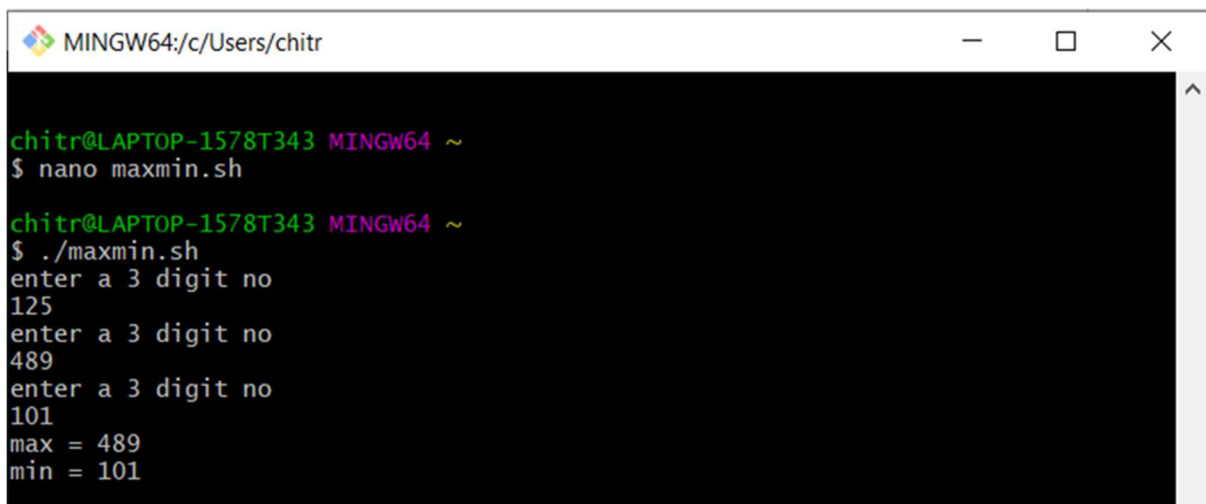
PROGRAM:



```
MINGW64:/c/Users/chitr
GNU nano 4.9.3 maxmin.sh
#!/bin/bash
max=0
min=999
for (( i=1; i<4; i++ ))
do
    echo $"enter a 3 digit no"
    read n$i
    if [ $((n$i)) -gt $max ];
    then
        max=$((n$i));
    fi
    if [ $((n$i)) -lt $min ];
    then
        min=$((n$i));
    fi
done
echo $"max = $((max))"
echo $"min = $((min))"

[ Read 18 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell ^_ Go To Line
```

PROGRAM OUTPUT



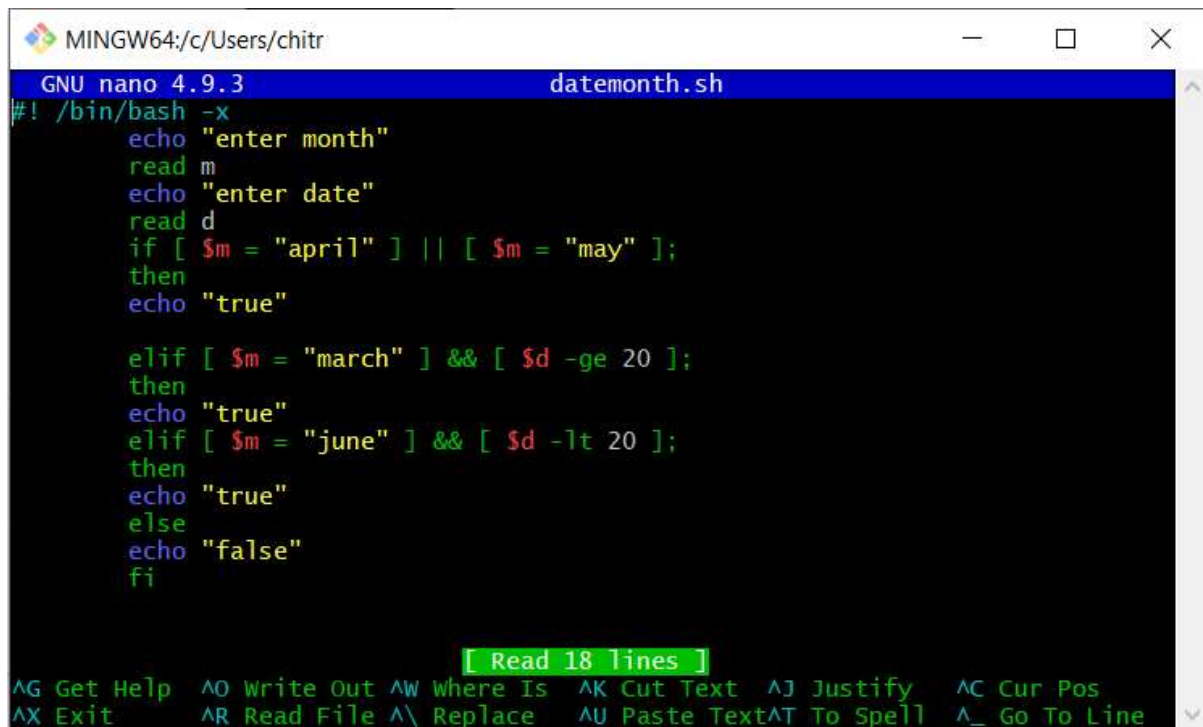
```
MINGW64:/c/Users/chitr
chitr@LAPTOP-1578T343 MINGW64 ~
$ nano maxmin.sh

chitr@LAPTOP-1578T343 MINGW64 ~
$ ./maxmin.sh
enter a 3 digit no
125
enter a 3 digit no
489
enter a 3 digit no
101
max = 489
min = 101
```

2. 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.

```
chitr@LAPTOP-1578T343 MINGW64 ~  
$ nano datemonth.sh
```

PROGRAM:



```
MINGW64:/c/Users/chitr  
GNU nano 4.9.3 datemonth.sh  
#!/bin/bash -x  
  echo "enter month"  
  read m  
  echo "enter date"  
  read d  
  if [ $m = "april" ] || [ $m = "may" ];  
  then  
    echo "true"  
  
    elif [ $m = "march" ] && [ $d -ge 20 ];  
    then  
      echo "true"  
    elif [ $m = "june" ] && [ $d -lt 20 ];  
    then  
      echo "true"  
    else  
      echo "false"  
    fi
```

[Read 18 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~  
$ ./datemonth.sh  
enter month  
june  
enter date  
20  
false
```

3. 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.

```
chitr@LAPTOP-1578T343 MINGW64 ~  
$ nano leapyear.sh
```

PROGRAM:

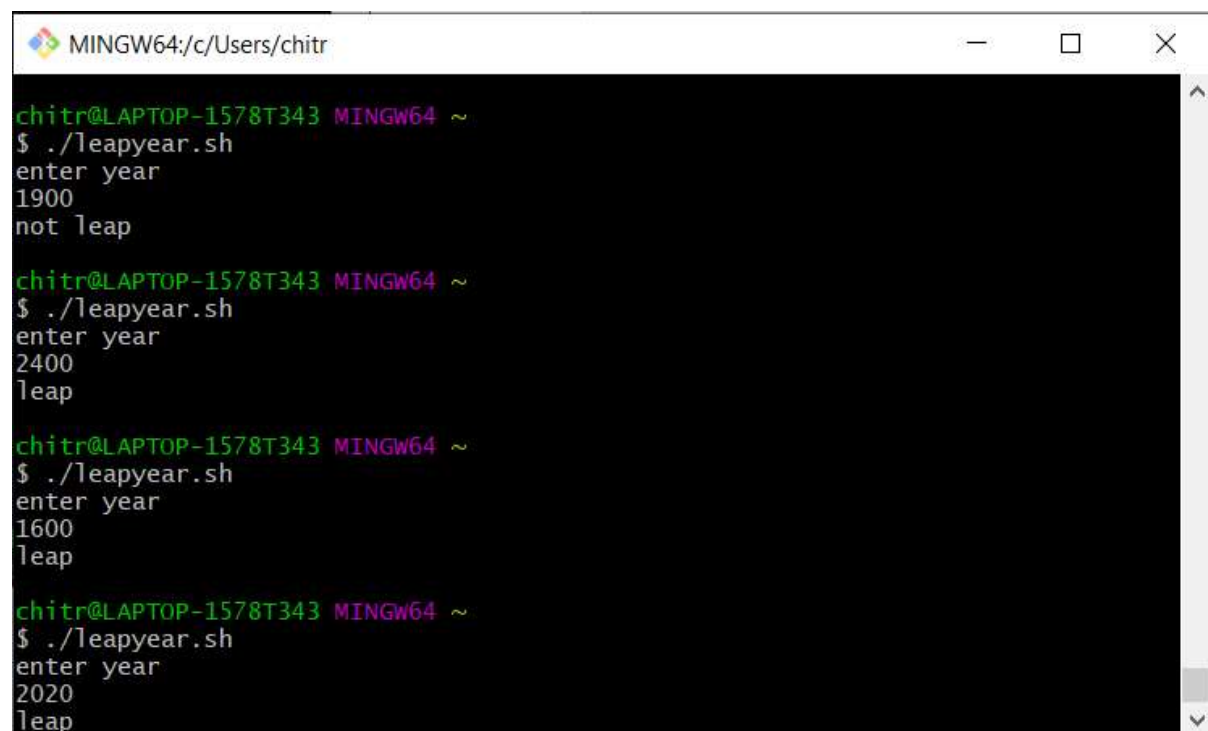


The screenshot shows a terminal window titled 'MINGW64:/c/Users/chitr' with a nano editor open to a file named 'leapyear.sh'. The editor's title bar indicates 'GNU nano 4.9.3'. The script content is as follows:

```
#!/bin/bash  
echo "enter year"  
read year  
rem1=$((year%4));  
rem2=$((year%100));  
rem3=$((year%400));  
if [ $rem1 -eq 0 ] && [ $rem2 -ne 0 ];  
then  
echo "leap"  
elif [ $rem3 -eq 0 ];  
then  
echo "leap"  
else  
echo "not leap"  
fi
```

At the bottom of the editor, a status bar shows '[Read 15 lines]' and a list of keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^\ Replace, ^U Paste Text, ^T To Spell, ^_ Go To Line.

OUTPUT:

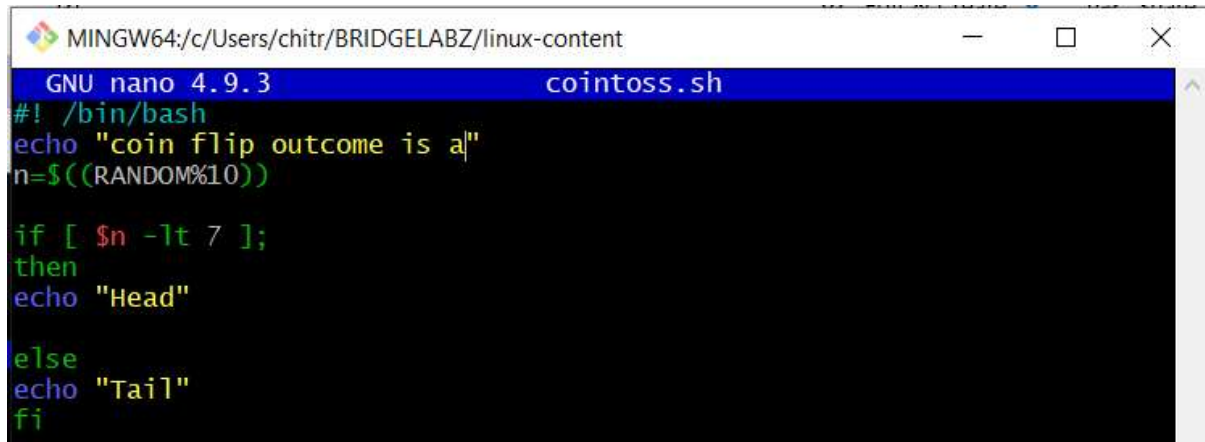


The screenshot shows a terminal window titled 'MINGW64:/c/Users/chitr' with the following output from running the script:

```
chitr@LAPTOP-1578T343 MINGW64 ~  
$ ./leapyear.sh  
enter year  
1900  
not leap  
  
chitr@LAPTOP-1578T343 MINGW64 ~  
$ ./leapyear.sh  
enter year  
2400  
leap  
  
chitr@LAPTOP-1578T343 MINGW64 ~  
$ ./leapyear.sh  
enter year  
1600  
leap  
  
chitr@LAPTOP-1578T343 MINGW64 ~  
$ ./leapyear.sh  
enter year  
2020  
leap
```

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

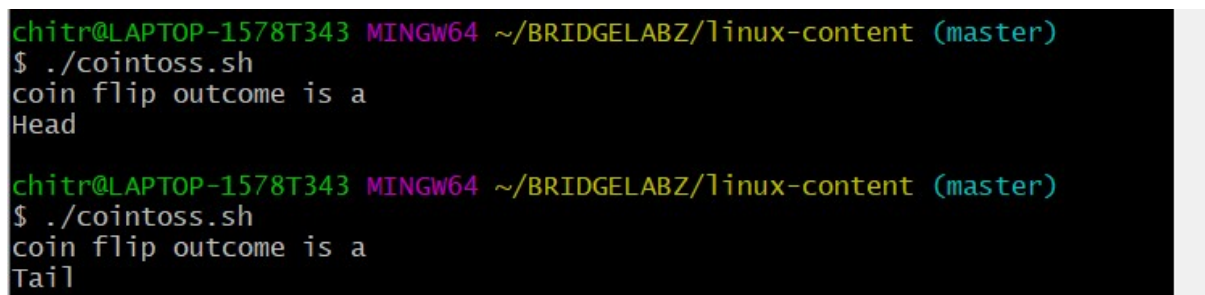
PROGRAM:



```
MINGW64:/c/Users/chitr/BRIDGELABZ/linux-content
GNU nano 4.9.3 cointoss.sh
#!/bin/bash
echo "coin flip outcome is a"
n=$((RANDOM%10))

if [ $n -lt 7 ];
then
echo "Head"
else
echo "Tail"
fi
```

OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./cointoss.sh
coin flip outcome is a
Head

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./cointoss.sh
coin flip outcome is a
Tail
```

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

PROGRAM:


```
MINGW64:/c/Users/chitr
GNU nano 4.9.3 numinwords.sh Modified
#!/bin/bash
echo "enter a sinle digit no"
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 "You have not entered a single digit no"

fi

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~
$ ./numinwords.sh
enter a sinle digit no
7
SEVEN
```

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

PROGRAM:


```
MINGW64:/c/Users/chitr
GNU nano 4.9.3 weekday.sh
#!/bin/bash
echo "enter a sinle digit no between 1 to 7"
read n

if [ $n -eq 1 ];
then
echo "Sunday"

elif [ $n -eq 2 ];
then
echo "Monday"

elif [ $n -eq 3 ];
then
echo "Tuesday"

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 "enter a valid digit"

fi

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~
$ ./numinwords.sh
enter a sinle digit no
7
SEVEN
```

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

```
chitr@LAPTOP-1578T343 MINGW64 ~  
$ nano 10multiples.sh
```

PROGRAM:

```
MINGW64:/c/Users/chitr
GNU nano 4.9.3 10multiples.sh
#!/bin/bash
echo "enter a sinle digit no"
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 "lakh"

elif [ $n -eq 1000000 ];
then
echo "ten lakh"

elif [ $n -eq 10000000 ];
then
echo "crore"

elif [ $n -eq 100000000 ];
then
echo "ten crore"

else
echo "enter a valid number either one or a multiple of 10"
fi

AG Get Help  AO Write Out  AW Where Is  AK Cut Text  AJ Justify  AC Cur Pos
AX Exit      AR Read File  A\ Replace  AU Paste TextAT To Spell  ^_ Go To Line
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~  
$ ./10multiples.sh  
enter a sinle digit no  
100  
hundred
```

4. Enter 3 Numbers do following arithmetic operation and find the one that is maximum & minimum 1. $a + b * c$ 2. $a \% b + c$ 3. $c + a / b$ 4. $a * b + c$

```
chitr@LAPTOP-1578T343 MINGW64 ~  
$ nano aritmaticopmaxmin.sh
```

PROGRAM:

```
GNU nano 4.9.3 aritmaticopmaxmin.sh Modified  
#!/bin/bash  
echo "Enter number a"  
read a  
echo "Enter number b"  
read b  
echo "Enter number c"  
read c  
  
p=$((a+b*c))  
q=$((a%b+c))  
r=$((c+a*b))  
s=$((a*b+c))  
  
echo "a+b*c = $p"  
echo "a%b+c = $q"  
echo "c+a*b = $r"  
echo "a*b+c = $s"  
  
if [ $p -ge $q ] && [ $p -ge $r ] && [ $p -ge $s ];  
then  
echo "max is a+b*c=$p"  
elif [ $q -ge $p ] && [ $q -ge $r ] && [ $q -ge $s ];  
then  
echo "max is a%b+c=$q"  
elif [ $r -ge $p ] && [ $r -ge $q ] && [ $r -ge $s ];  
then  
echo "max is c+a*b=$r"  
elif [ $s -ge $p ] && [ $s -ge $q ] && [ $s -ge $r ];  
then  
echo "max is a*b+c=$s"  
fi  
  
if [ $p -le $q ] && [ $p -le $r ] && [ $p -le $s ];  
then  
echo "min is a+b*c=$p"  
elif [ $q -le $p ] && [ $q -le $r ] && [ $q -le $s ];  
then  
echo "min is a%b+c=$q"  
elif [ $r -le $p ] && [ $r -le $q ] && [ $r -le $s ];  
then  
echo "min is c+a/b=$r"  
elif [ $s -le $p ] && [ $s -le $q ] && [ $s -le $r ];  
then  
echo "min is a*b+c=$s"  
fi
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~
$ ./aritmaticopmaxmin.sh
Enter number a
5
Enter number b
4
Enter number c
3
a+b*c = 17
a%b+c = 4
c+a*b = 23
a*b+c = 23
max is c+a*b=23
min is a%b+c=4
```

CASE STATEMENTS

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

PROGRAM:

```
GNU nano 4.9.3 numtoword.sh
#!/bin/bash

echo "Enter a single digit number"
read n
case $n in
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 "enter a single digit no only" ;;
esac
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./numtoword.sh
Enter a single digit number
7
seven
```

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

PPROGRAM:

```
MINGW64:/c/Users/chitr/BRIDGELABZ/linux-content
GNU nano 4.9.3 numtoday.sh
#!/bin/bash

echo "Enter a number in the range 1 to 7"
read n
case $n in
1)
echo "sunday" ;;
2)
echo "monday" ;;
3)
echo "tuesday" ;;
4)
echo "wednesday" ;;
5)
echo "thursday" ;;
6)
echo "friday" ;;
7)
echo "saturday" ;;
*)
echo "enter a valid number" ;;
esac
```

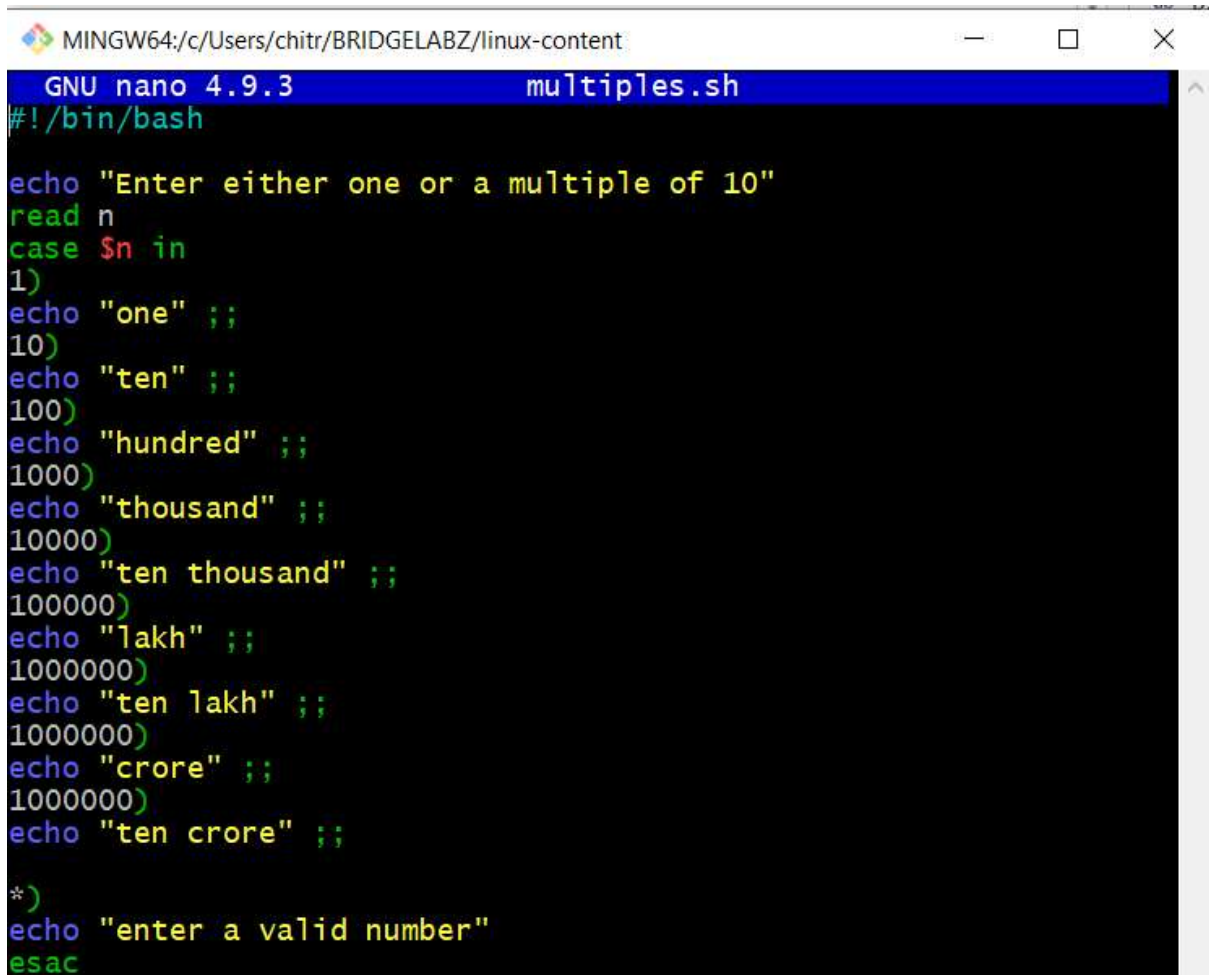
OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./numtoday.sh
Enter a number in the range 1 to 7
5
thursday
```


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

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ nano multiples.sh
```

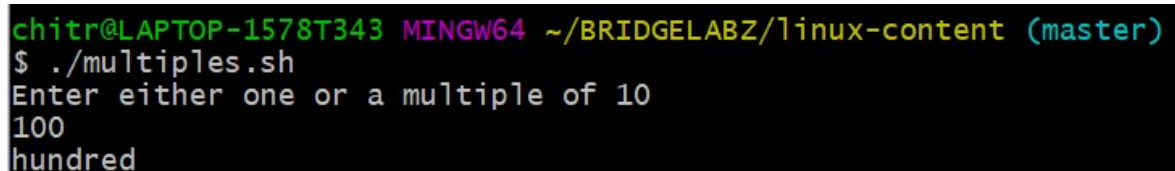
PROGRAM:



```
MINGW64:/c/Users/chitr/BRIDGELABZ/linux-content
GNU nano 4.9.3 multiples.sh
#!/bin/bash

echo "Enter either one or a multiple of 10"
read n
case $n in
1)
echo "one" ;;
10)
echo "ten" ;;
100)
echo "hundred" ;;
1000)
echo "thousand" ;;
10000)
echo "ten thousand" ;;
100000)
echo "lakh" ;;
1000000)
echo "ten lakh" ;;
10000000)
echo "crore" ;;
100000000)
echo "ten crore" ;;
*)
echo "enter a valid number"
esac
```

OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./multiples.sh
Enter either one or a multiple of 10
100
hundred
```

4. Write a program that takes User Inputs and does Unit Conversion of different Length units1. Feet to Inch 2. Feet to Meter 3. Inch to Feet 4. Meter to Feet

PROGRAM:

```
MINGW64:/c/Users/chitr/BRIDGELABZ/linux-content
GNU nano 4.9.3 conversions.sh
#!/bin/bash
echo "Enter a number"
echo "1 for feet to inch"
echo "2 for feet to meter"
echo "3 for inch to feet"
echo "4 for meter to feet"
read n

echo "Enter length"
read len

case $n in
1)
echo "feet=$len"
echo "inch = "
printf %.2f\\n "$((10000*len*12))e-4" ;;
2)
echo "feet=$len and"
echo "meter = "
printf %.2f\\n "$((10000*len*12.54/100))e-4" ;;
3)
echo "inch=$len"
echo "feet = "
printf %.2f\\n "$((10000*len/12))e-4" ;;
4)
echo "meter=$len"
echo "inch = "
printf %.2f\\n "$((10000*len*100/12.54))e-4" ;;
*)
echo "enter a valid no" ;;
esac
```

OUTPUT:

```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ/linux-content (master)
$ ./conversions.sh
Enter a number
1 for feet to inch
2 for feet to meter
3 for inch to feet
4 for meter to feet
3
Enter length
100
inch=100
feet =
8.33
```