

## Assignment No.02

**Name:** - Omprakash Khawshi

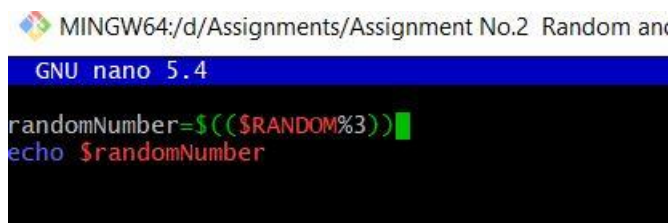
\*\*\*\*\*

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

**RANDOM** is a shell variable that is used to generate random integers in Linux. It is an internal bash command that returns a pseudo-random 16-bit integer in the range 0 – 32767. It returns a different integer at each invocation.

**Code:** -

```
randomNumber=$((RANDOM%3))  
echo $randomNumber
```



The screenshot shows a terminal window titled 'MINGW64:/d/Assignments/Assignment No.2 Random and if & else'. The prompt is 'GNU nano 5.4'. The code entered is:  
randomNumber=\$((RANDOM%3))  
echo \$randomNumber

**Output:** -



The screenshot shows a terminal window titled 'Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else'. The prompt is '\$ ./Q1.sh'. The output of the script is shown for five consecutive runs:  
0  
2  
1  
1  
2

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

Code: -

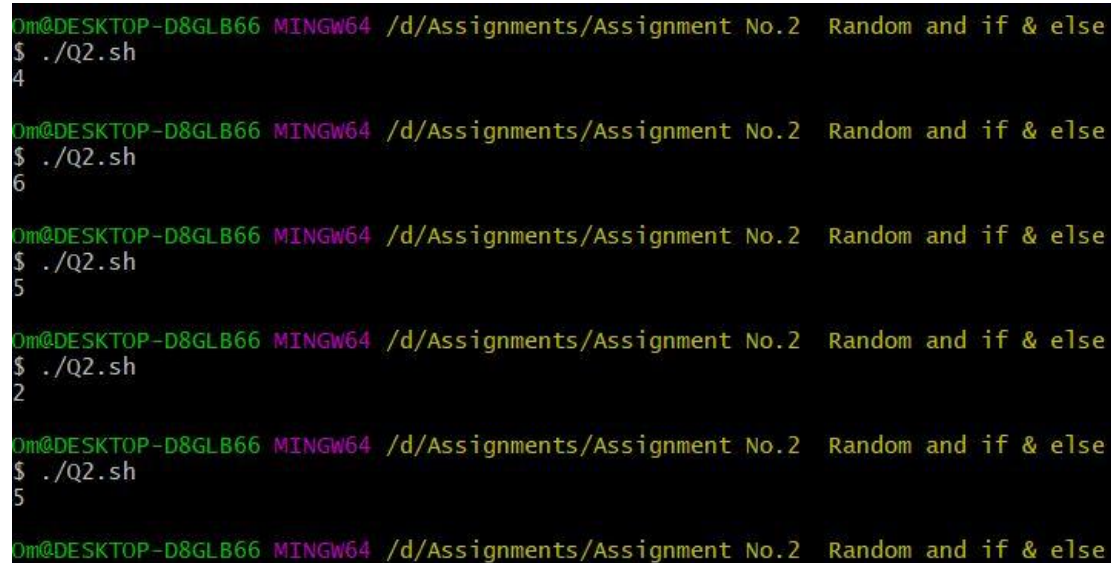
```
randomNumber=$((1+$RANDOM%6))  
echo $randomNumber
```

 MINGW64:/d/Assignments/Assignment No.2 Random and if & else



```
GNU nano 5.4 Q2.sh  
randomNumber=$((1+$RANDOM%6))  
echo $randomNumber
```

Output: -




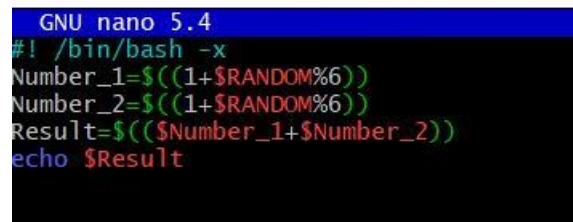
```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else  
$ ./Q2.sh  
4  
  
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else  
$ ./Q2.sh  
6  
  
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else  
$ ./Q2.sh  
5  
  
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else  
$ ./Q2.sh  
2  
  
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else  
$ ./Q2.sh  
5  
  
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
```

### Q.3 Add two Random Dice Number and Print the Result.

Code: -

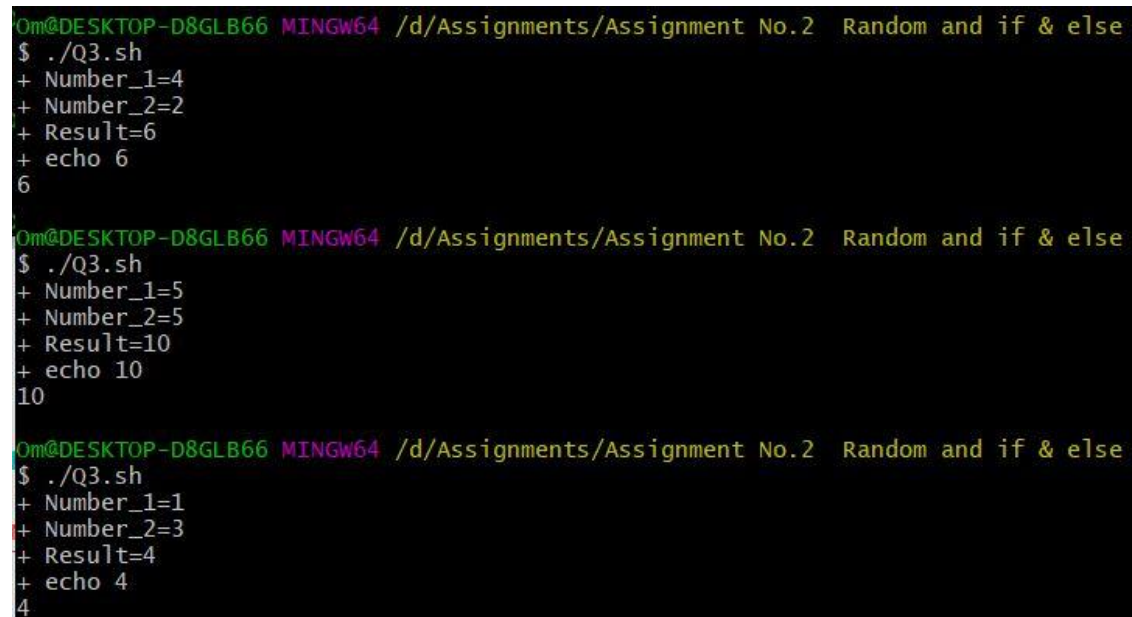
```
#!/bin/bash -x
Number_1=$((1+$RANDOM%6))
Number_2=$((1+$RANDOM%6))
Result=$((Number_1+Number_2))
echo $Result
```

 MINGW64:/d/Assignments/Assignment No.2 Rar



```
GNU nano 5.4
#!/bin/bash -x
Number_1=$((1+$RANDOM%6))
Number_2=$((1+$RANDOM%6))
Result=$((Number_1+Number_2))
echo $Result
```

Output: -



```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q3.sh
+ Number_1=4
+ Number_2=2
+ Result=6
+ echo 6
6

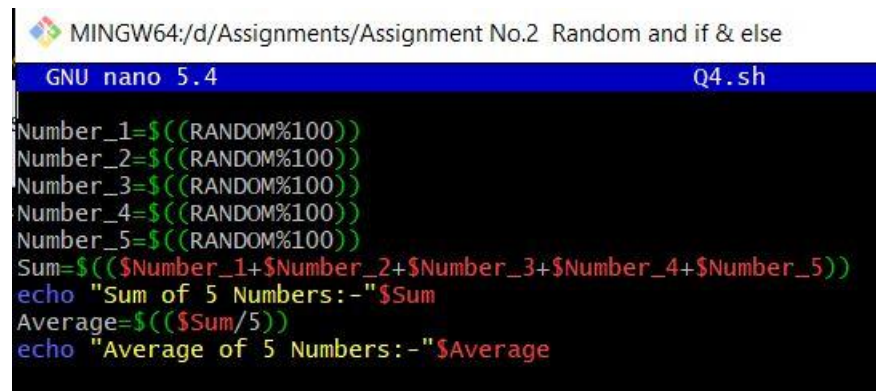
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q3.sh
+ Number_1=5
+ Number_2=5
+ Result=10
+ echo 10
10

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q3.sh
+ Number_1=1
+ Number_2=3
+ Result=4
+ echo 4
4
```

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

**Code: -**

```
Number_1=$((RANDOM%100))
Number_2=$((RANDOM%100))
Number_3=$((RANDOM%100))
Number_4=$((RANDOM%100))
Number_5=$((RANDOM%100))
Sum=$((Number_1+Number_2+Number_3+Number_4+Number_5))
echo "Sum of 5 Numbers:-"$Sum
Average=$((Sum/5))
echo "Average of 5 Numbers:-"$Average
```



The screenshot shows a terminal window with the title bar "MINGW64:/d/Assignments/Assignment No.2 Random and if & else". The window contains the GNU nano 5.4 editor editing a file named "Q4.sh". The script content is as follows:

```
Number_1=$((RANDOM%100))
Number_2=$((RANDOM%100))
Number_3=$((RANDOM%100))
Number_4=$((RANDOM%100))
Number_5=$((RANDOM%100))
Sum=$((Number_1+Number_2+Number_3+Number_4+Number_5))
echo "Sum of 5 Numbers:-"$Sum
Average=$((Sum/5))
echo "Average of 5 Numbers:-"$Average
```

## Output: -

```
MINGW64:/d/Assignments/Assignment No.2 Random and if & else
$ ./Q4.sh
+ Number_1=42
+ Number_2=5
+ Number_3=99
+ Number_4=0
+ Number_5=36
+ Sum=182
+ echo 'Sum of 5 Numbers:-182'
Sum of 5 Numbers:-182
+ Average=36
+ echo 'Average of 5 Numbers:-36'
Average of 5 Numbers:-36

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ nano Q4.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q4.sh
Sum of 5 Numbers:-292
Average of 5 Numbers:-58

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q4.sh
Sum of 5 Numbers:-218
Average of 5 Numbers:-43

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q4.sh
Sum of 5 Numbers:-291
Average of 5 Numbers:-58

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q4.sh
Sum of 5 Numbers:-191
Average of 5 Numbers:-38
```

## Q.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 acre

Code: -

```
echo -ne " 1.Inches to feets\n 2.Rectangular Plot of 60 feet x 40 feet in meters \n Enter Your Choice:- "
```

```
read Num
```

```
case $Num in
```

```
1)
```

```
    echo " Enter the Inches:- "
```

```
    read a
```

```
    f2i=$((a /12))
```

```
    echo " Number of feet:- "$f2i
```

```
    ;;
```

```
2)
```

```
    echo " Enter Rectangular Height :- "
```

```
    read Height
```

```
    echo " Enter Rectangular Width :- "
```

```
    read Width
```

```
    Rectangular_Area=$(expr $Width \* $Height)
```

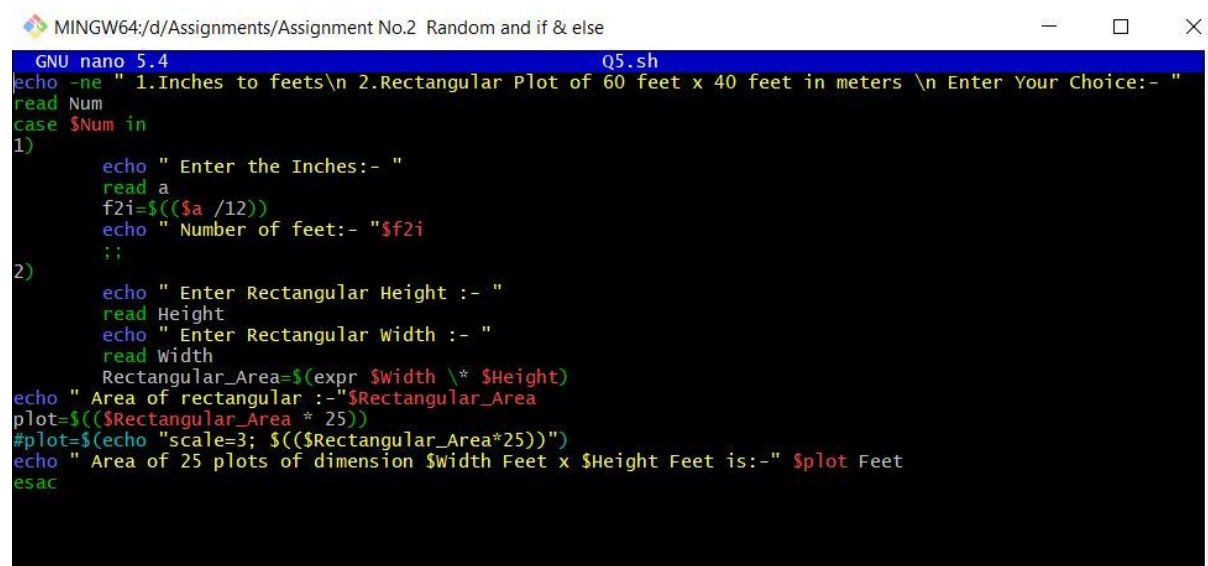
```
echo " Area of rectangular :-"$Rectangular_Area
```

```
plot=$((Rectangular_Area * 25))
```

```
#plot=$(echo "scale=3; $((Rectangular_Area*25))")
```

```
echo " Area of 25 plots of dimension $Width Feet x $Height Feet is:-" $plot Feet
```

```
esac
```



```
MINGW64/d/Assignments/Assignment No.2 Random and if & else
GNU nano 5.4 Q5.sh
echo -ne " 1.Inches to feets\n 2.Rectangular Plot of 60 feet x 40 feet in meters \n Enter Your Choice:- "
read Num
case $Num in
1)
    echo " Enter the Inches:- "
    read a
    f2i=$((a /12))
    echo " Number of feet:- "$f2i
    ;;
2)
    echo " Enter Rectangular Height :- "
    read Height
    echo " Enter Rectangular Width :- "
    read Width
    Rectangular_Area=$(expr $Width \* $Height)
echo " Area of rectangular :-"$Rectangular_Area
plot=$((Rectangular_Area * 25))
#plot=$(echo "scale=3; $((Rectangular_Area*25))")
echo " Area of 25 plots of dimension $Width Feet x $Height Feet is:-" $plot Feet
esac
```

## Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q5.sh
1.Inches to feets
2.Rectangular Plot of 60 feet x 40 feet in meters
Enter Your Choice:- 1
Enter the Inches:-
42
Number of feet:- 3

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ ./Q5.sh
1.Inches to feets
2.Rectangular Plot of 60 feet x 40 feet in meters
Enter Your Choice:- 2
Enter Rectangular Height :-
5
Enter Rectangular Width :-
2
Area of rectangular :-10
Area of 25 plots of dimension 2 Feet x 5 Feet is:- 250 Feet

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else
$ |
```

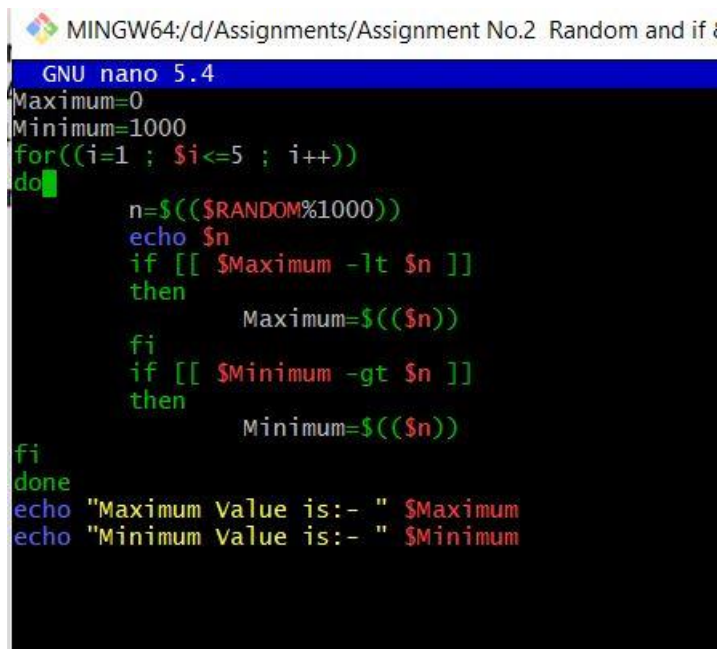


## If Else

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

**Code: -**

```
Maximum=0
Minimum=1000
for((i=1 ; $i<=5 ; i++))
do
    n=$(($RANDOM%1000))
    echo $n
    if [[ $Maximum -lt $n ]]
    then
        Maximum=$((n))
    fi
    if [[ $Minimum -gt $n ]]
    then
        Minimum=$((n))
    fi
done
echo "Maximum Value is:- " $Maximum
echo "Minimum Value is:- " $Minimum
```



```
MINGW64:/d/Assignments/Assignment No.2 Random and if
GNU nano 5.4
Maximum=0
Minimum=1000
for((i=1 ; $i<=5 ; i++))
do
    n=$(($RANDOM%1000))
    echo $n
    if [[ $Maximum -lt $n ]]
    then
        Maximum=$((n))
    fi
    if [[ $Minimum -gt $n ]]
    then
        Minimum=$((n))
    fi
done
echo "Maximum Value is:- " $Maximum
echo "Minimum Value is:- " $Minimum
```



## Output: -

MINGW64/d/Assignments/Assignment No.2 Random and if & else/if & else/Q.1 Write a program that reads 5 Random 3 Digit values and then outputs the minimum

```
nm@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.1 Write a program that reads 5 Random 3
t values and then outputs the minimum
$ ./Q1.sh
685
809
796
154
532
Maximum Value is:- 796
Minimum Value is:- 154


nm@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.1 Write a program that reads 5 Random 3
t values and then outputs the minimum
$ ./Q1.sh
661
849
955
651
6
Maximum Value is:- 955
Minimum Value is:- 6


nm@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.1 Write a program that reads 5 Random 3
t values and then outputs the minimum
$ |
```

**Q.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.**

**Code: -**

```
#!/bin/bash
read -p "Enter Date :-" Date
read -p "Emter Month :-" Month
if [[ $Month -eq "march" && $Date -gt 20 && $Date -lt 31 ]]
then echo "True"
elif [[ $Month -eq "april" && $Date -lt 30 ]]
then echo "True"
elif [[ $Month -eq "may" && $Date -lt 31 ]]
then echo "True"
elif [[ $Month -eq "june" && $Date -lt 20 ]]
then echo "True"
else
echo "False"
fi
```

 MINGW64:/d/Assignments/Assignment No.2 Random and if & else/if & else/Q.2 Write a program that take



```
GNU nano 5.4 Q2.sh
#!/bin/bash
read -p "Enter Date :-" Date
read -p "Emter Month :-" Month
if [[ $Month -eq "march" && $Date -gt 20 && $Date -lt 31 ]]
then echo "True"
elif [[ $Month -eq "april" && $Date -lt 30 ]]
then echo "True"
elif [[ $Month -eq "may" && $Date -lt 31 ]]
then echo "True"
elif [[ $Month -eq "june" && $Date -lt 20 ]]
then echo "True"
else
echo "False"
fi
```

## Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.2 Write a program that takes day and month from the command line and prints true if
$ ./Q2.sh
Enter Date :-14
Enter Month :-march
True

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.2 Write a program that takes day and month from the command line and prints true if
$ ./Q2.sh
Enter Date :-21
Enter Month :-june
True

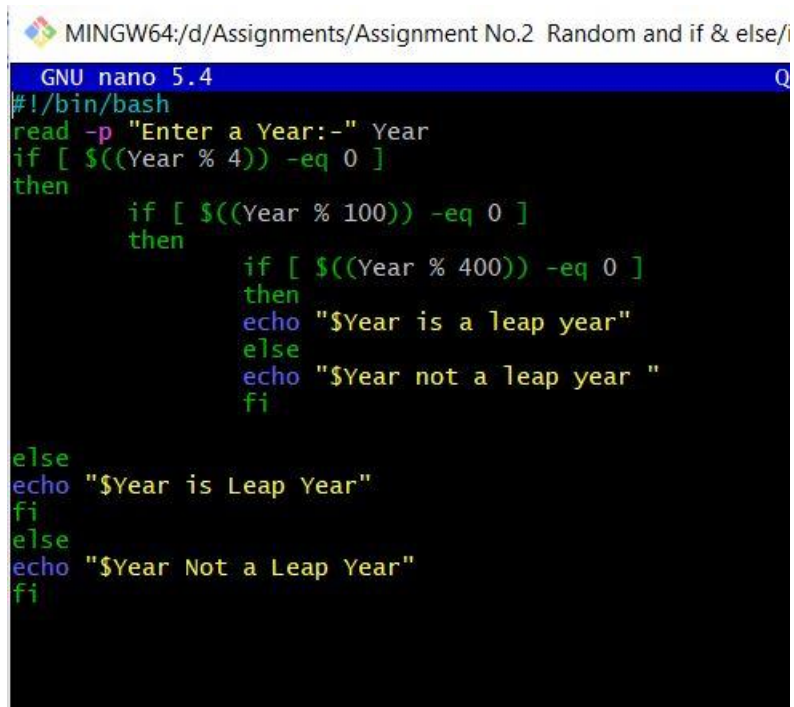
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.2 Write a program that takes day and month from the command line and prints true if
$ ./Q2.sh
Enter Date :-31
Enter Month :-april
False

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.2 Write a program that takes day and month from the command line and prints true if
$ |
```

**Q.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.**

**Code: -**

```
#!/bin/bash
read -p "Enter a Year:-" Year
if [ $((Year % 4)) -eq 0 ]
then
    if [ $((Year % 100)) -eq 0 ]
    then
        if [ $((Year % 400)) -eq 0 ]
        then
            echo "$Year is a leap year"
        else
            echo "$Year not a leap year "
        fi
    else
        echo "$Year is Leap Year"
    fi
else
    echo "$Year Not a Leap Year"
fi
```

A screenshot of a terminal window titled 'MINGW64:/d/Assignments/Assignment No.2 Random and if & else/i'. The terminal shows the GNU nano 5.4 editor with the same Bash script as above. The script prompts the user to 'Enter a Year:-' and then checks if the year is a leap year based on the specified conditions. The script is displayed in a dark-themed editor with syntax highlighting: keywords like 'if', 'then', 'else', 'fi', and 'echo' are in green, and variables and operators are in yellow/white. The terminal window has a standard Linux-style title bar and a small icon on the left.

## Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.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 a 4 Digit Number, Divisible by 4 and not 100 unless divisible by 400
$ ./Q3.sh
Enter a Year:-1997
1997 Not a Leap Year

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.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 a 4 Digit Number, Divisible by 4 and not 100 unless divisible by 400
$ ./Q3.sh
Enter a Year:-2016
2016 is Leap Year

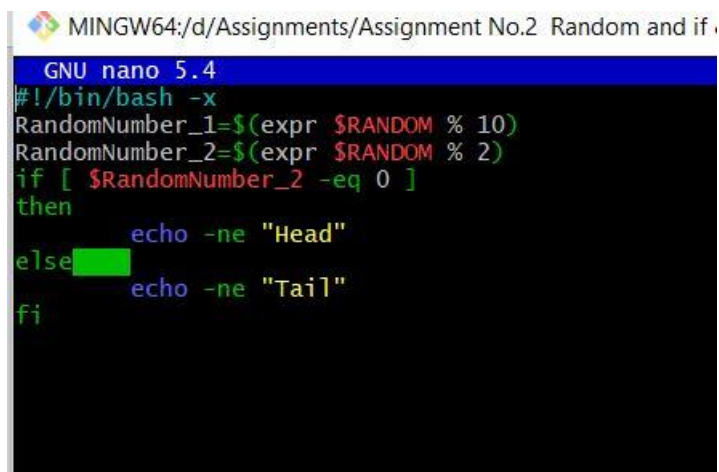
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.3 Write a program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year. A Leap Year checks for a 4 Digit Number, Divisible by 4 and not 100 unless divisible by 400
$ ./Q3.sh
Enter a Year:-2014
2014 Not a Leap Year

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.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 a 4 Digit Number, Divisible by 4 and not 100 unless divisible by 400
$ ./Q3.sh
Enter a Year:-2021
2021 Not a Leap Year
```

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

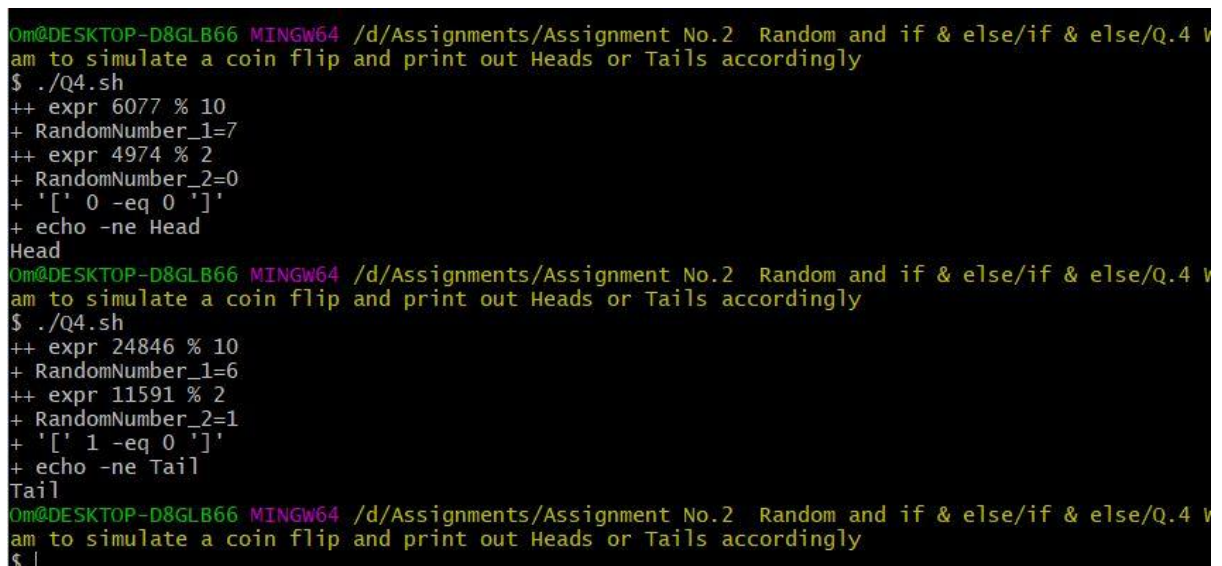
**Code: -**

```
#!/bin/bash -x
RandomNumber_1=$(expr $RANDOM % 10)
RandomNumber_2=$(expr $RANDOM % 2)
if [ $RandomNumber_2 -eq 0 ]
then
    echo -ne "Head"
else
    echo -ne "Tail"
fi
```

A screenshot of a terminal window titled 'MINGW64:/d/Assignments/Assignment No.2 Random and if'. The terminal shows the GNU nano 5.4 editor with the following code:

```
#!/bin/bash -x
RandomNumber_1=$(expr $RANDOM % 10)
RandomNumber_2=$(expr $RANDOM % 2)
if [ $RandomNumber_2 -eq 0 ]
then
    echo -ne "Head"
else
    echo -ne "Tail"
fi
```

**Output: -**

A screenshot of a terminal window showing the execution of the script. The prompt is 'Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.4 W'. The user enters 'am to simulate a coin flip and print out Heads or Tails accordingly' and then './Q4.sh'. The output shows the script's execution steps, including random number generation and the final output 'Head'. The user then runs the script again, and the output shows 'Tail'.

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.4 W
am to simulate a coin flip and print out Heads or Tails accordingly
$ ./Q4.sh
++ expr 6077 % 10
+ RandomNumber_1=7
++ expr 4974 % 2
+ RandomNumber_2=0
+ '[' 0 -eq 0 ']'
+ echo -ne Head
Head
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.4 W
am to simulate a coin flip and print out Heads or Tails accordingly
$ ./Q4.sh
++ expr 24846 % 10
+ RandomNumber_1=6
++ expr 11591 % 2
+ RandomNumber_2=1
+ '[' 1 -eq 0 ']'
+ echo -ne Tail
Tail
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if & else/Q.4 W
am to simulate a coin flip and print out Heads or Tails accordingly
$ |
```

## if, elif and else

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

**Code: -**

```
read -p "Enter a Number between 1 to 9 :-" Number
if(($Number ==1))
then
    echo "One";
elif(($Number ==2))
then
    echo "$Number Two"
elif(($Number ==3))
then
    echo "$Number Three"
elif(($Number ==4))
then
    echo "$Number Four"
elif(($Number ==5))
then
    echo "$Number Five"
elif(($Number ==6))
then
    echo "$Number Six"
elif(($Number ==7))
then
    echo "$Number Seven"
elif(($Number ==8))
then
    echo "$Number Eight"
elif(($Number ==9))
then
    echo "$Number Nine"
else
    echo "Enter Number between 1 to 9 "
fi
```



MINGW64:/d/Assignments/Assignment No.2 Random and if & else/if, e

```
GNU nano 5.4
read -p "Enter a Number between 1 to 9 :-" Number
if(($Number ==1))
then
    echo "One";
elif(($Number ==2))
then
    echo "$Number Two"
elif(($Number ==3))
then
    echo "$Number Three"
elif(($Number ==4))
then
    echo "$Number Four"
elif(($Number ==5))
then
    echo "$Number Five"
elif(($Number ==6))
then
    echo "$Number Six"
elif(($Number ==7))
then
    echo "$Number Seven"
elif(($Number ==8))
then
    echo "$Number Eight"
elif(($Number ==9))
then
    echo "$Number Nine"
else
    echo "Enter Number between 1 to 9 "
fi
```

**Output: -**

MINGW64:/d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q.1 Read a single digit nur

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, e
a single digit number and write the number in word
$ ./Q1.sh
Enter a Number between 1 to 9 :-4
4 Four

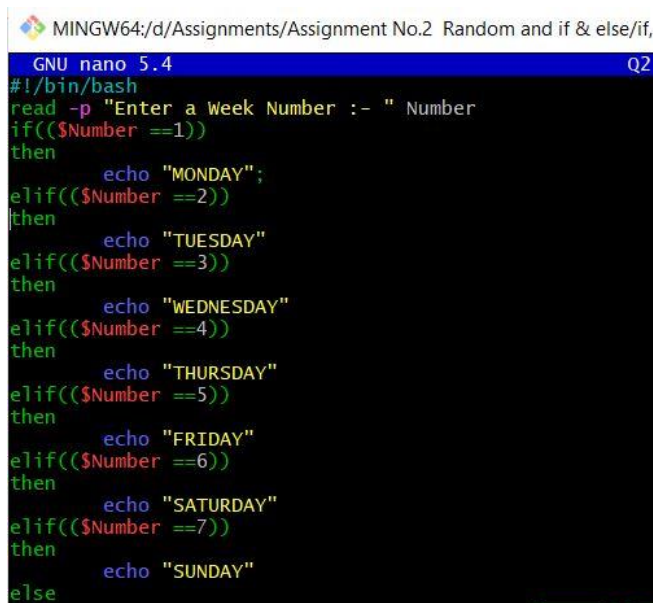
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, e
a single digit number and write the number in word
$ ./Q1.sh
Enter a Number between 1 to 9 :-5
5 Five

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, e
a single digit number and write the number in word
$ ./Q1.sh
Enter a Number between 1 to 9 :-8
8 Eight
```

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


**Code: -**

```
#!/bin/bash
read -p "Enter a Week Number :- " Number
if(($Number ==1))
then
    echo "MONDAY";
elif(($Number ==2))
then
    echo "TUESDAY"
elif(($Number ==3))
then
    echo "WEDNESDAY"
elif(($Number ==4))
then
    echo "THURSDAY"
elif(($Number ==5))
then
    echo "FRIDAY"
elif(($Number ==6))
then
    echo "SATURDAY"
elif(($Number ==7))
then
    echo "SUNDAY"
else
```



The screenshot shows a terminal window titled "MINGW64/d/Assignments/Assignment No.2 Random and if & else/if,". The terminal is running GNU nano 5.4. The code displayed is the same shell script as shown in the previous block, with the following lines visible: `#!/bin/bash`, `read -p "Enter a Week Number :- " Number`, `if(($Number ==1))`, `then`, `echo "MONDAY";`, `elif(($Number ==2))`, `then`, `echo "TUESDAY"`, `elif(($Number ==3))`, `then`, `echo "WEDNESDAY"`, `elif(($Number ==4))`, `then`, `echo "THURSDAY"`, `elif(($Number ==5))`, `then`, `echo "FRIDAY"`, `elif(($Number ==6))`, `then`, `echo "SATURDAY"`, `elif(($Number ==7))`, `then`, `echo "SUNDAY"`, and `else`. The terminal has a dark background with light-colored text.

## Output: -

 MINGW64:/d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q.2 Read a Numb

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & els
a Number and Display the week day (Sunday, Monday,...)
$ ./Q2.sh
Enter a Week Number :- 1
MONDAY

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & els
a Number and Display the week day (Sunday, Monday,...)
$ ./Q2.sh
Enter a Week Number :- 2
TUESDAY

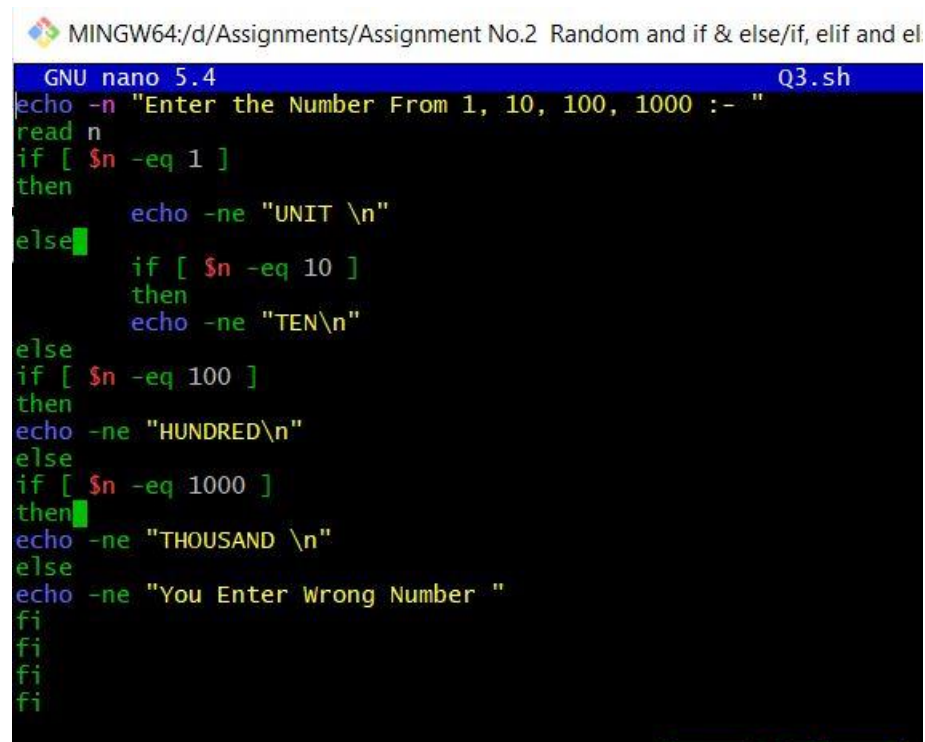
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & els
a Number and Display the week day (Sunday, Monday,...)
$ ./Q2.sh
Enter a Week Number :- 3
WEDNESDAY

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & els
a Number and Display the week day (Sunday, Monday,...)
$ ./Q2.sh
Enter a Week Number :- 4
THURSDAY
```

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

**Code: -**

```
echo -n "Enter the Number From 1, 10, 100, 1000 :- "  
read n  
if [ $n -eq 1 ]  
then  
    echo -ne "UNIT \n"  
else  
    if [ $n -eq 10 ]  
    then  
        echo -ne "TEN\n"  
    else  
        if [ $n -eq 100 ]  
        then  
            echo -ne "HUNDRED\n"  
        else  
            if [ $n -eq 1000 ]  
            then  
                echo -ne "THOUSAND \n"  
            else  
                echo -ne "You Enter Wrong Number "  
            fi  
        fi  
    fi  
fi
```



```
MINGW64:/d/Assignments/Assignment No.2 Random and if & else/if, elif and el  
GNU nano 5.4 Q3.sh  
echo -n "Enter the Number From 1, 10, 100, 1000 :- "  
read n  
if [ $n -eq 1 ]  
then  
    echo -ne "UNIT \n"  
else  
    if [ $n -eq 10 ]  
    then  
        echo -ne "TEN\n"  
    else  
        if [ $n -eq 100 ]  
        then  
            echo -ne "HUNDRED\n"  
        else  
            if [ $n -eq 1000 ]  
            then  
                echo -ne "THOUSAND \n"  
            else  
                echo -ne "You Enter Wrong Number "  
            fi  
        fi  
    fi  
fi
```

## Output: -

```
MINGW64:/d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q.3 Read a Number 1, 10, 100, 1... [
$ nano Q3.sh
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q
a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,
$ ./Q3.sh
Enter the Number From 1, 10, 100, 1000 :- 10
TEN

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q
a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,
$ ./Q3.sh
Enter the Number From 1, 10, 100, 1000 :- 1
UNIT

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q
a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,
$ ./Q3.sh
Enter the Number From 1, 10, 100, 1000 :- 100
HUNDRED

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q
a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,
$ ./Q3.sh
Enter the Number From 1, 10, 100, 1000 :- 12
You Enter Wrong Number
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q
a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,
$ |
```

**Q.4 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.  $a * b + c$**

**Code: -**

```
read -p "Enter a First Number:- " a
read -p "Enter a Second Number:- " b
read -p "Enter a Third Number:- " c
A=$(( a + b * c ))
B=$(( a % b + c ))
C=$(( c + a / b ))
D=$(( a * b + c ))
echo $A
echo $B
echo $C
echo $D
if (( $A > $B && $A > $C && $A > $D ))
then
echo "$A is Maximum "
elif (( $B > $A && $B > $C && $B > $D ))
then
echo "$B is Maximum "
elif (( $C > $A && $C > $B && $C > $D ))
then
echo "$C is Maximum"
else
echo "$D is Maximum"
fi
if (( $A < $B && $A < $C && $A < $D ))
then
echo "$A is Minimum "
elif (( $B < $A && $B < $C && $B < $D ))
then
echo "$B is Minimum"
elif (( $C < $A && $C < $B && $C < $D ))
then
echo "$C is Minimum"
else
echo "$D is Minimum"
```



fi

```
read -p "Enter a First Number:- " a
read -p "Enter a Second Number:- " b
read -p "Enter a Third Number:- " c
A=$(( a + b * c ))
B=$(( a % b + c ))
C=$(( c + a / b ))
D=$(( a * b + c ))
echo $A
echo $B
echo $C
echo $D
if (( $A > $B && $A > $C && $A > $D ))
then
echo "$A is Maximum "
elif (( $B > $A && $B > $C && $B > $D ))
then
echo "$B is Maximum "
elif (( $C > $A && $C > $B && $C > $D ))
then
echo "$C is Maximum"
else
echo "$D is Maximum"
fi
if (( $A < $B && $A < $C && $A < $D ))
then
echo "$A is Minimum "
elif (( $B < $A && $B < $C && $B < $D ))
then
echo "$B is Minimum"
elif (( $C < $A && $C < $B && $C < $D ))
then
echo "$C is Minimum"
else
echo "$D is Minimum"
fi
```

Output: -

```
MINGW64:/d/Assignments/Assignment No.2 Random and if & else/if, elif and else/Q.4 Enter 3 Number
3 Numbers do following arithmetic operation and find the one that is maximum and minimum
$ ./Q4.sh
Enter a First Number:- 2
Enter a Second Number:- 5
Enter a Third Number:- 1
7
3
1
11
11 is Maximum
1 is Minimum

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else,
3 Numbers do following arithmetic operation and find the one that is maximum and minimum
$ ./Q4.sh
Enter a First Number:- 5
Enter a Second Number:- 8
Enter a Third Number:- 77
621
82
77
117
621 is Maximum
77 is Minimum
```



## case statement

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

**Code: -**


```
read -p "Enter Single Digit Number:- " number
case $number 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"
;;
esac
```

```

read -p "Enter Single Digit Number:- " number
case $number 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"
;;
esac

```

### Output:-

 MINGW64:/d/Assignments/Assignment No.2 Random and if & else/case statement/Q.1 Read a single digit numt

```

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case s
ingle digit number and write the number in word using Case
$ ./Q1.sh
Enter Single Digit Number:- 5
Five

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case s
ingle digit number and write the number in word using Case
$ ./Q1.sh
Enter Single Digit Number:- 7
Seven

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case s
ingle digit number and write the number in word using Case
$ ./Q1.sh
Enter Single Digit Number:- 1
One

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case s
ingle digit number and write the number in word using Case
$ ./Q1.sh
Enter Single Digit Number:- 0
Zero

```

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

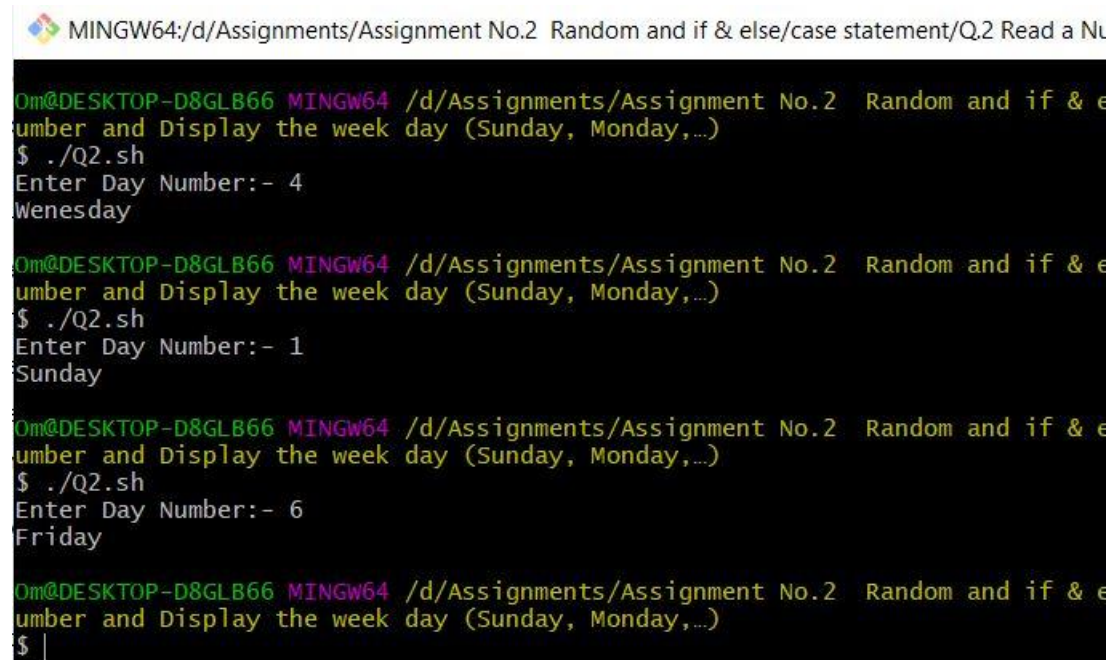
**Code: -**

```
read -p "Enter Day Number:- " number
case $number in
1)
echo "Sunday"
;;
2)
echo "Monday"
;;
3)
echo "Thesday"
;;
4)
echo "Wenesday"
;;
5)
echo "Thursday"
;;
6)
echo "Friday"
;;
7)
echo "Saturday"
esac
```



```
read -p "Enter Day Number:- " number
case $number in
1)
echo "Sunday"
;;
2)
echo "Monday"
;;
3)
echo "Thesday"
;;
4)
echo "Wenesday"
;;
5)
echo "Thursday"
;;
6)
echo "Friday"
;;
7)
echo "Saturday"
esac
```

## Output: -



The screenshot shows a Windows command prompt window with the title bar "MINGW64:/d/Assignments/Assignment No.2 Random and if & else/case statement/Q.2 Read a Number and Display the week day (Sunday, Monday,...)". The prompt is "Om@DESKTOP-D8GLB66 MINGW64". The user enters the command `./Q2.sh`, and the script prompts "Enter Day Number:-". The user enters "4", and the script outputs "Wenesday". The user then enters "1", and the script outputs "Sunday". The user then enters "6", and the script outputs "Friday". Finally, the user enters a pipe character "|", and the prompt returns.

```
MINGW64:/d/Assignments/Assignment No.2 Random and if & else/case statement/Q.2 Read a Number and Display the week day (Sunday, Monday,...)
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.2 Read a Number and Display the week day (Sunday, Monday,...)
$ ./Q2.sh
Enter Day Number:- 4
Wenesday

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.2 Read a Number and Display the week day (Sunday, Monday,...)
$ ./Q2.sh
Enter Day Number:- 1
Sunday

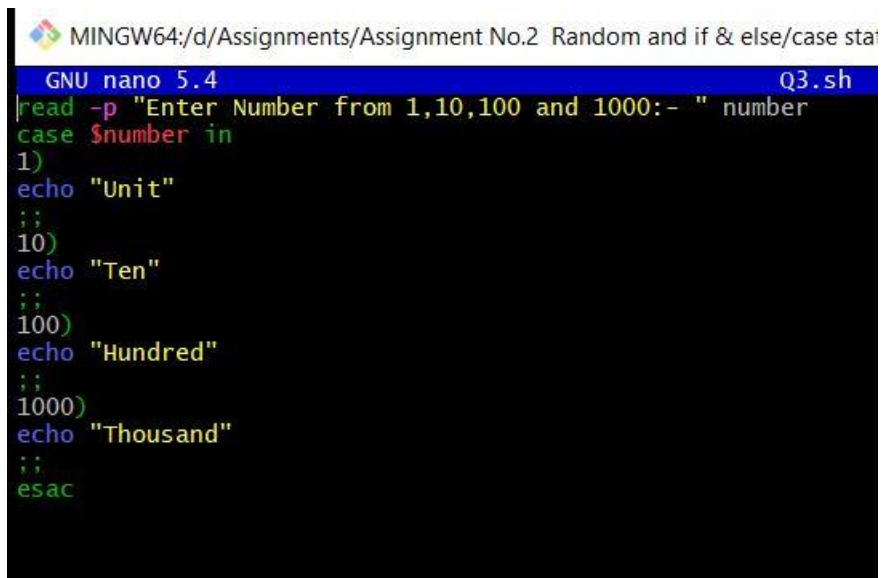
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.2 Read a Number and Display the week day (Sunday, Monday,...)
$ ./Q2.sh
Enter Day Number:- 6
Friday

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.2 Read a Number and Display the week day (Sunday, Monday,...)
$ |
```

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

**Code: -**

```
read -p "Enter Number from 1,10,100 and 1000:- " number
case $number in
1)
echo "Unit"
;;
10)
echo "Ten"
;;
100)
echo "Hundred"
;;
1000)
echo "Thousand"
;;
esac
```



```
MINGW64:/d/Assignments/Assignment No.2 Random and if & else/case sta
GNU nano 5.4 Q3.sh
read -p "Enter Number from 1,10,100 and 1000:- " number
case $number in
1)
echo "Unit"
;;
10)
echo "Ten"
;;
100)
echo "Hundred"
;;
1000)
echo "Thousand"
;;
esac
```

## Output: -

```
MINGW64:/d/Assignments/Assignment No.2 Random and if & else/case statement/Q.3 Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,....

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.3 Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,....
$ ./Q3.sh
Enter Number from 1,10,100 and 1000:- 1
Unit

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.3 Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,....
$ ./Q3.sh
Enter Number from 1,10,100 and 1000:- 10
Ten

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.3 Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,....
$ ./Q3.sh
Enter Number from 1,10,100 and 1000:- 100
Hundred

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement/Q.3 Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,....
$ ./Q3.sh
Enter Number from 1,10,100 and 1000:- 1000
Thousand
```

**Q.4 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**

**Code: -**

```
echo -ne "1. Feet to Inch \n2. Inch to feet \n3. Feet Into Meter \n4. Meter into Feet \nEnter
Your Choice :- "
read Number
case $Number in
1)
read -p "Enter Feets :- " Number
Inch=$(( $Number * 12 ))
echo "Number of Inches $Inch "
;;
2)
read -p "Enter Inches :- " Number
feet=$(( $Number / 12 ))

echo "Number of feets $feet"
;;
3)
read -p "Enter Feets :- " Number
meter=$(( $Number / 3 ))
echo "Number of Meter $meter "
;;
4)
read -p "Enter Meters :- " Number
feet=$(( $Number * 3 ))
echo "Number of Feets $feet "
;;
esac
```



MINGW64; d/Assignments/Assignment No.2 Random

```
GNU nano 5.4
echo -ne "1. Feet to Inch \n2. Inch to feet\n3. Feet Into Meter\n4. Meter into Feet\nEnter Your Choice :- "
read Number
case $Number in
1)
read -p "Enter Feets :- " Number
Inch=$(( $Number * 12 ))
echo "Number of Inches $Inch "
;;
2)
read -p "Enter Inches :- " Number
feet=$(( $Number / 12 ))
echo "Number of feets $feet"
;;
3)
read -p "Enter Feets :- " Number
meter=$(( $Number / 3 ))
echo "Number of Meter $meter "
;;
4)
read -p "Enter Meters :- " Number
feet=$(( $Number * 3 ))
echo "Number of Feets $feet "
;;
esac
```

## Output: -

MINGW64; d/Assignments/Assignment No.2 Random and if & else/case statement/Q.4 Write a program that takes User Input

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement
program that takes User Inputs and does Unit Conversion of different Length units
$ ./Q4.sh
1. Feet to Inch
2. Inch to feet
3. Feet Into Meter
4. Meter into Feet
Enter Your Choice :- 1
Enter Feets :- 44
Number of Inches 528

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement
program that takes User Inputs and does Unit Conversion of different Length units
$ ./Q4.sh
1. Feet to Inch
2. Inch to feet
3. Feet Into Meter
4. Meter into Feet
Enter Your Choice :- 2
Enter Inches :- 40
Number of feets 3

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement
program that takes User Inputs and does Unit Conversion of different Length units
$ ./Q4.sh
1. Feet to Inch
2. Inch to feet
3. Feet Into Meter
4. Meter into Feet
Enter Your Choice :- 3
Enter Feets :- 99
Number of Meter 33

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.2 Random and if & else/case statement
program that takes User Inputs and does Unit Conversion of different Length units
$ ./Q4.sh
1. Feet to Inch
2. Inch to feet
3. Feet Into Meter
4. Meter into Feet
Enter Your Choice :- 4
Enter Meters :- 33
Number of Feets 99
```