Day-05 Assignment Solutions (Submitted By- **Rupesh Roy**)

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

Code: echo $(($RANDOM % 9 + 1))

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ echo $(( $RANDOM % 9 + 1 ))
8

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ echo $(( $RANDOM % 9 + 1 ))
5

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ echo $(( $RANDOM % 9 + 1 ))
2
```

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

Code: echo $(($RANDOM % 6 + 1))

OutPut:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ echo $(( $RANDOM % 6 + 1 ))
6

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ echo $(( $RANDOM % 6 + 1 ))
4

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ echo $(( $RANDOM % 6 + 1 ))
2
```

3. Add two Random Dice Number and Print the Result

Code:

```
#!/bin/bash
Dice1=$(( $RANDOM % 6 + 1 ))
Dice2=$(( $RANDOM % 6 + 1 ))
result=$(( $Dice1 + $Dice2 ))
echo "Addition of two random dice number is:" $result
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./RandomDiceAdd.sh
Addition of two random dice number is: 7

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./RandomDiceAdd.sh
Addition of two random dice number is: 6

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./RandomDiceAdd.sh
Addition of two random dice number is: 5
```

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

Code:

```
#! /bin/bash
sum=0
for ((count=1 ; count<=5 ; count++))
do
   TwoDigitRandomNo=$(( RANDOM % 9 + 1))$(( RANDOM % 10 ))
    sum=$(( $sum + $TwoDigitRandomNo ))
done
        echo "Sum of 5 random 2 digit no is:" $sum
        echo "Average is:" $(( $sum/5 ))
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./SumRandomTwoDigit.sh
Sum of 5 random 2 digit no is: 334
Average is: 66

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./SumRandomTwoDigit.sh
Sum of 5 random 2 digit no is: 290
Average is: 58
```

5. Unit Conversion

A. 1ft = 12 in then 42 in = ? ft

Code:
```
#!/bin/bash
echo "Unit Conversion inch into feet"
read -p "Enter inch:" inch
printf %.2f "$((1000000000 * ($inch*1)/12))e-9"
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./UnitConversion.sh
Unit Conversion inch into feet
Enter inch:42
3.50
```

B. Rectangular Plot of 60 feet x 40 feet in meters-

Code:
```
#!/bin/bash
echo "Dimension of square field is 60ft x 40ft."
echo "Area in feet:"
Dimension=$((60 * 40))
echo $Dimension "sqr feet"
echo "Area in meter:"
printf %.3f "$((1000000000 * ($Dimension*9290)/100000))e-9"
echo " sqr meter"
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./UnitConversion.sh
Dimension of square field is 60ft x 40ft.
Area in feet:
2400 sqr feet
Area in meter:
222.960 sqr meter
```

C. Calculate area of 25 such plots in acres-

Code:
```
#!/bin/bash
echo "Dimension of 25 square field is 60ft x 40ft."
echo "Area in feet:"
Dimension=$((25 * 60 * 40))
echo $Dimension "sqr feet"
echo "Area in Acres:"
printf %.3f "$((1000000000 * ($Dimension*1)/43560))e-9"
```

```
echo " sqr Acre"
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./Area_inAcres.sh
Dimension of 25 square field is 60ft x 40ft.
Area in feet:
60000 sqr feet
Area in meter:
1.377 sqr meter
```

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

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 Day:" day
read -p "Enter Month:" month
if [[ $day -ge 20 && $day -le 31 && $month -ge 3 && $month -le 6 ]]
then
        echo "True"
else
        echo "False"
fi
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./DayMonth.sh
Enter Day:24
Enter Month:4
True

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./DayMonth.sh
Enter Month:6
True

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./DayMonth.sh
Enter Day:12
Enter Month:2
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.

Code:

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

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./LeapYear.sh
Enter Year:2000
2000 is leap year

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./LeapYear.sh
Enter Year:1999
1999 is not leap year

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./LeapYear.sh
Enter Year:1998
1998 is not a leap year
```

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

Code:

```
#!/bin/bash
isHead=1
randomCheck=$(( RANDOM%2 ))
echo "Coin is Fliping..."
if [ $isHead -eq $randomCheck ]
  then
    echo "HEAD"
  else
     echo "TAIL"
fi
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./Coin_Flip.sh
Coin is Fliping...
HEAD

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./Coin_Flip.sh
Coin is Fliping...
TAIL
```

Selection Practice Problems with if, elif and else:

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

Code:

```
#!/bin/bash
read -p "Enter single digit no:" number
if [ $number -eq 0 ]
then
        echo $number"-Zero"
elif [ $number -eq 1 ]
then
        echo $number"-One"
elif [ $number -eq 2 ]
  then
        echo $number"-Two"
elif [ $number -eq 3 ]
 then
        echo $number"-Three"
elif [ $number -eq 4 ]
  then
        echo $number"-Four"
elif [ $number -eq 5 ]
then
        echo $number"-Five"
elif [ $number -eq 6 ]
 then
        echo $number"-Six"
elif [ $number -eq 7 ]
 then
        echo $number"-Seven"
elif [ $number -eq 8 ]
then
        echo $number"-Eight"
else
        echo $number"-Nine"
fi
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./if_else_1.sh
Enter single digit no:4
4-Four

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./if_else_1.sh
Enter single digit no:6
6-Six

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./if_else_1.sh
Enter single digit no:9
9-Nine
```

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

Code:

```
#!/bin/bash
read -p "Enter number of Day:" number
if [ $number -eq 1 ]
        then
        echo "Day" $number "is MONDAY"
elif [ $number -eq 2 ]
         then
        echo "Day" $number "is TUESDAY"
elif [ $number -eq 3 ]
          then
         echo "Day" $number "is WEDNESDAY"
elif [ $number -eq 4 ]
        then
        $number "is FRIDAY"
elif [ $number -eq 6 ]
        then
        echo "Day" $number "is SATURDAY"
elif [ $number -eq 7 ]
        then
        echo "Day" $number "is SUNDAY"
else
        echo "Day no is not exist"
fi
```

Output:

```
$ ./weekDay.sh
Enter number of Day:4
Day 4 is THURSDAY

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./weekDay.sh
Enter number of Day:8
Day no is not exist

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./weekDay.sh
Enter number of Day:5
Day 5 is FRIDAY
```

Selection Practice Problems with case statement:

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

Code:

```
#!/bin/bash
read -p "Enter number=" number
case $number in
        0)echo $number"-ZERO"
        ;;
        1)echo $number"-ONE"
        ;;
        2)echo $number"-TWO"
        ;;
        3)echo $number"-THREE"
        ;;
        4)echo $number"-FOUR"
        ;;
        5)echo $number"-FIVE"
        ;;
        6)echo $number"-SIX"
        ;;
        7)echo $number"-SEVEN"
        ;;
        8)echo $number"-EIGHT"
        ;;
        9)echo $number"-NINE"
        ;;
        *) echo "Warning: only singal digit accepted."
esac
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./math.sh
Enter number=4
4-FOUR

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./math.sh
Enter number=9
9-NINE
```

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

Code:

```
#!/bin/bash
read -p "Enter Day number=" Day
case $Day in
        1)echo "Day is MONDAY"
        ;;
        2)echo "Day is TUESDAY"
        ;;
        3)echo "Day is WEDNESDAY"
        ;;
        4)echo "Day is THURSDAY"
        ;;
        5)echo "Day is FRIDAY"
        ;;
        6)echo "Day is SATURDAY"
        ;;
        7)echo "Day is SUNDAY"
        ;;
        *) echo "Day number is not EXIST"
esac
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
Enter Day number=3
Day is WEDNESDAY

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./math.sh
Enter Day number=6
Day is SATURDAY

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./math.sh
Enter Day number=8
Day number is not EXIST
```

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

Code:

```
#!/bin/bash
read -p "Enter number=" number
case $number in
        1)echo "Unit"
        ;;
        10)echo "Ten"
        ;;
        100)echo "Hundred"
        ;;
        1000)echo "Thousand"
        ;;
        10000)echo "Ten Thousand"
        ;;
        100000)echo "Hundred Thusand"
        ;;
        1000000)echo "Million"
        ;;
        *) echo " Number is not EXIST"
esac
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./math.sh
Enter number=1000000
Million

ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./math.sh
Enter number=10000
Ten Thousand
```

4. 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**

Code:

```
#!/bin/bash
        echo "1. Feet to Inch."
        echo "2. Feet to Meter."
        echo "3. Inch to Feet."
        echo "4. Meter to Feet."
        read -p "Enter Choice=" Choice
        case $Choice in
                1)echo "1. Feet to Inch."
                read -p "Enter number in feet:" feet
                 printf %.3f "$(($feet*12))"
                 echo " Inch"
                 ;;
                2)echo "2. Feet to Meter."
                read -p "Enter number in feet:" feet
                 printf %.3f "$((1000000000 * ($feet*3048)/10000))e-9"
                 echo " Meter"
                 ;;
                3)echo "3. Inch to Feet."
                read -p "Enter number in Inch:" Inch
                printf %.3f "$((1000000000 * ($Inch*1)/12))e-9"
                echo " Feet"
                ;;
                4)echo "4. Meter to Feet."
                read -p "Enter number in Meter" Meter
                printf %.3f "$((1000000000 * ($Meter*10000)/3048))e-9"
                echo " Feet"
                 ;;
                *) echo "Wrong Choice"
        esac
```

Output:

```
ROY@ROY-PC MINGW64 ~/Desktop/DeskTop/shellPrograming
$ ./CaseUnitConversion.sh
1. Feet to Inch.
2. Feet to Meter.
3. Inch to Feet.
Enter Choice=2
2. Feet to Meter.
Enter number in feet:2
0.610 Meter
```