

ASSIGNMENT-05

Selection Practice Problems with if & else

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

```
Prudhvi@PrudhviReddy MINGW64 /e/BridgeLabz/Assignmen
$ ./3digitmin_maxvalues.sh
Random Generated numbers are : 497 382 586 788 183
The maximum number is 788
The minimum number is 183
```

```
#!/bin/bash

arr=();
for ((i=1; i<=5; i++))
do
    num=$(( $RANDOM%900+100 ))
    arr+=($num)
done

echo "Random Generated numbers are : ${arr[@]}"

max=${arr[0]}
min=${arr[0]}

for i in ${arr[@]}
do
    if [[ $i -gt $max ]]
    then
        max="$i"
    fi

    if [[ $i -lt $min ]]
    then
        min="$i"
    fi
done

echo "The maximum number is $max"
echo "The minimum number is $min"
```

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.

```
#!/bin/bash -x

read -p "Enter Date: " date
read -p "Enter Month: " Month

# should validate input: non-blank, numeric

combo=0
res="false"
st=1

if [ $Month -ge 3 -a $Month -le 6 ]; then
    # 30 for Apr, Jun
    # 31 for Mar, May
    dlimit=$((30 + (Month % 2)))
    if [ $date -ge 1 -a $date -le $dlimit ]; then
        combo=$((Month * 100 + date))
        # true Mar 20 to Jun 20
        if [ $combo -ge 320 -a $combo -le 620 ]; then
            res="true"
            st=0
        fi
    fi
fi

echo "$Month/$date $res"
exit $st
```

```

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./datewithinspecificrange.sh
Enter Date: 3
Enter Month: 6
6/3 true

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./datewithinspecificrange.sh
Enter Date: 25
Enter Month: 6
6/25 false

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./datewithinspecificrange.sh
Enter Date: 25
Enter Month: 5
5/25 true

```

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 40.

```

#!/bin/bash -x

read -p "Enter year: " year

if [ $(( year % 4 )) -eq 0 ]
then
    if [ $(( year % 100 )) -eq 0 ]
    then
        if [ $((year%400)) -eq 0 ]
        then
            echo "$year is leap."
        else
            echo "$year is not leap"
        fi
    else
        echo "$year is leap"
    fi
else
    echo "$year is not leap."
fi

```

```

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./leapyear.sh
Enter year: 2000
2000 is leap.

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./leapyear.sh
Enter year: 1696
1696 is leap

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./leapyear.sh
Enter year: 3005
3005 is not leap.

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./leapyear.sh
Enter year: 2020
2020 is leap

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/Prob02
$ ./leapyear.sh
Enter year: 2019
2019 is not leap.

```

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

```

#!/bin/bash

for ((i=0;i>=0;i++))
do
    if [[ $((RANDOM%2)) = 1 ]]
    then
        echo Head
    else
        echo Tail
    fi
done

```

```

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02
$ ./head_tail.sh
Tail
Head
Tail
Tail
Tail
Tail
Tail
Tail
Head

```

Selection Practice Problems with if, elif and else

Read a single digit number and write the number in word

```
#!/bin/bash

read -p "Enter Number : " num

len=${#num}

for (( i=1; i<=$len; i++ ))
do
    digit=$(( echo $num | cut -c $i ))

    if [[ $digit = 0 ]]
    then
        echo zero
    elif [[ $digit = 1 ]]
    then
        echo one
    elif [[ $digit = 2 ]]
    then
        echo two
    elif [[ $digit = 3 ]]
    then
        echo three
    elif [[ $digit = 4 ]]
    then
        echo four
    elif [[ $digit = 5 ]]
    then
        echo five
    elif [[ $digit = 6 ]]
    then
        echo six
    elif [[ $digit = 7 ]]
    then
        echo seven
    elif [[ $digit = 8 ]]
    then
        echo eight
    elif [[ $digit = 9 ]]
    then
        echo nine
    fi
done
```

```
Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintnum.sh
Enter Number : 123
one
two
three

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintnum.sh
Enter Number : 12
one
two

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintnum.sh
Enter Number : 5
five
```

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

```
Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintday.sh
Enter Number : 12
sunday
monday

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintday.sh
Enter Number : 5
thursday

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintday.sh
Enter Number : 7
saturday
```

```
#!/bin/bash

read -p "Enter Number : " num

len=${#num}

for (( i=1; i<=$len; i++ ))
do
    day=$( echo $num | cut -c $i )

    if [[ $day = 1 ]]
    then
        echo sunday
    elif [[ $day = 2 ]]
    then
        echo monday
    elif [[ $day = 3 ]]
    then
        echo tuesday
    elif [[ $day = 4 ]]
    then
        echo wednesday
    elif [[ $day = 5 ]]
    then
        echo thursday
    elif [[ $day = 6 ]]
    then
        echo friday
    elif [[ $day = 7 ]]
    then
        echo saturday
    else
        echo there are no more days than seven in a week
    fi
done
```

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

```
Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintunit.sh
Enter Number : 369
963
units : 9
ten's : 6
Hundereds : 3

Prudhvi@PrudhviReddy MINGW64 /e/Bridgelabz/Assignments/Day5/prob02/ifndef
$ ./readnumprintunit.sh
Enter Number : 9863
3689
units : 3
ten's : 6
Hundereds : 8
Thousands : 9
```

```
#!/bin/bash

read -p "Enter Number : " n

len=${#n}
#echo $len
for (( i=$len - 1; i>= 0; i-- ))
do
    rev=$rev${n:$i:1}
done
echo $rev;
num=$rev;

for (( i=1; i<=$len; i++ ))
do
    unit=$(( echo $num | cut -c $i ))

    if [[ $i = 1 ]]
    then
        echo "units : "$unit
    elif [[ $i = 2 ]]
    then
        echo "ten's : "$unit
    elif [[ $i = 3 ]]
    then
        echo "Hundereds :"$unit
    elif [[ $i = 4 ]]
    then
        echo "Thousands :"$unit
    elif [[ $i = 5 ]]
    then
        echo "Ten Thousands :"$unit
    fi
done
```

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. $a * b + c$

Prudhvi@PrudhviReddy MINGW64

false

\$./minmaxarthoperations.sh

Enter 1st number : 1

Enter 2nd number : 2

Enter 3rd number : 3

$a+b*c$: 7

$a\%b+c$: 4

$c+a/b$: 3

$a*b+c$: 5

7 4 3 5

The maximum number is 7

The minimum number is 3

```
#!/bin/bash
arr=();
read -p "Enter 1st number : " a
read -p "Enter 2nd number : " b
read -p "Enter 3rd number : " c
one=$((a+b*c))
echo "a+b*c : "$one
arr+=($one)
two=$((a%b+c))
echo "a%b+c :"$two
arr+=($two)
three=$((c+a/c))
echo "c+a/b :"$three
arr+=($three)
four=$((a*b+c))
echo "a*b+c :"$four
arr+=($four)

echo ${arr[@]}

minimum=${arr[0]}
maximum=${arr[0]}

for i in ${arr[@]}
do
    if [[ $i -gt $maximum ]]
    then
        maximum="$i"
    fi
    if [[ $i -lt $minimum ]]
    then
        minimum="$i"
    fi
done
echo "The maximum number is $maximum"
echo "The minimum number is $minimum"
```

Selection Practice Problems with case statement

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

```
Prudhvi@PrudhviReddy MINGW6
ase
$ ./readnumprintword.sh
Enter Number : 2
two

Prudhvi@PrudhviReddy MINGW6
ase
$ ./readnumprintword.sh
Enter Number : 36
three
six

Prudhvi@PrudhviReddy MINGW6
ase
$ ./readnumprintword.sh
Enter Number : 369
three
six
nine
```

```
#!/bin/bash

read -p "Enter Number : " num

len=${#num}

for (( i=1; i<=$len; i++ ))
do
    digit=$( echo $num | cut -c $i )
case $digit 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
done
```

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

```
Prudhvi@PrudhviReddy MIN
ase
$ ./readnumprintday.sh
Enter Number : 2
monday

Prudhvi@PrudhviReddy MIN
ase
$ ./readnumprintday.sh
Enter Number : 36
tuesday
friday
```

```
#!/bin/bash

read -p "Enter Number : " num

len=${#num}

for (( i=1; i<=$len; i++ ))
do
    digit=$( echo $num | cut -c $i )
    case $digit in
        1)echo sunday;;
        2)echo monday;;
        3)echo tuesday;;
        4)echo wednesday;;
        5)echo thursday;;
        6)echo friday;;
        7)echo saturday;;
        *)echo no more daYS THAN 7 IN A WEEK;;
    esac
done
```

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

```
#!/bin/bash

read -p "Enter Number : " n

len=${#n}
for (( i=$len - 1; i>= 0; i-- ))
do
    rev=$rev${n:$i:1}
done
echo $rev;
num=$rev;

for (( i=1; i<=$len; i++ ))
do
    digit=$( echo $num | cut -c $i )
case $i in
1)echo "units :" $digit;;
2)echo "ten's :" $digit;;
3)echo "hundered's :" $digit;;
4)echo "Thousand's :" $digit;;
5)echo "Ten Thousand's :" $digit;;
esac
done
```

```
Prudhvi@PrudhviReddy MIN
ase
$ ./readnumprintunit.sh
Enter Number : 123
units : 3
ten's : 2
hundered's : 1

Prudhvi@PrudhviReddy MIN
ase
$ ./readnumprintunit.sh
Enter Number : 369
units : 9
ten's : 6
hundered's : 3
```

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

```
Prudhvi@PrudhviReddy MINGW64
ase
$ ./unitconversion.sh
enter inches :132
Converting inches to feet
11.00 ft
CONVERTING FEET TO METER
enter feet :12
3.65 mtr
converting feet to inch
enter feet :32
384 inch
converting meter to feet
enter meter :32
104.992 ft
```

```
#!/bin/bash

read -p "enter inches :" inch
echo Converting inches to feet
Feet=$(echo "scale=2;$inch*1/12" | bc)
echo $Feet ft

echo CONVERTING FEET TO METER

read -p "enter feet :" ft
meter=$(echo "scale=2;$ft/3.281" | bc)
echo $meter mtr

echo converting feet to inch

read -p "enter feet :" feet
in=$(echo "scale=2;$feet*12" | bc)
echo $in inch

echo converting meter to feet

read -p "enter meter :" mtr
ft=$(echo "scale=4;$mtr*3.281" | bc)
echo $ft ft
```