

SEQUENCES PRACTICE PROBLEMS

Ques 1. Use Random function to get single digit.

Solution :

```
$ echo $(( RANDOM % 10 ))  
4
```

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

Solution :

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

Ques 3. Add two Random Dice Number and print the result.

Solution :

```
$ cat Q3.sh  
#!/bin/bash -x  
a = $(( RANDOM % 6 + 1 ))  
b = $(( RANDOM % 6 + 1 ))  
c = $(( $a + $b ))  
echo "Sum : " $c
```

```
$ ./Q3.sh  
+ a=3  
+ b=5  
+ c=8  
+ echo 'Sum : ' 8  
Sum : 8
```

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

Solution :

```
$ cat Q4.sh  
#!/bin/bash -x  
sum=0  
for ((i=1; i<=5; i++))
```

```
do
    a=$((RANDOM%100))
    sum=$((sum+a))
done
avg=$((sum/5))
echo "Sum : "$sum
echo "Average : "$avg
```

\$./Q4.sh

```
+ sum=0
+ (( i=1 ))
+ (( i<=5 ))
+ a=44
+ sum=44
+ (( i++ ))
+ (( i<=5 ))
+ a=75
+ sum=119
+ (( i++ ))
+ (( i<=5 ))
+ a=94
+ sum=213
+ (( i++ ))
+ (( i<=5 ))
+ a=22
+ sum=235
+ (( i++ ))
+ (( i<=5 ))
+ a=97
+ sum=332
+ (( i++ ))
+ (( i<=5 ))
+ avg=66
+ echo 'Sum : 332'
Sum : 332
+ echo 'Average : 66'
Average : 66
```

Ques 5. Unit Conversion

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

Solution :

```
$ a=$((42 / 12))
```

```
$ echo $a
```

```
3.5
```

b. Rectangular plot of 60 feet x 40 feet in meters.

Solution :

```
$ cat feetInMeters.sh
```

```
#!/bin/bash -x
```

```
f=3.28084
```

```
a=$(echo 60 / $f | bc -l)
```

```
b=$(echo 40 / $f | bc -l)
```

```
echo $a" meters x "$b" meters "
```

```
$ ./feetInMeters.sh
```

```
+ f=3.28084
```

```
++ bc -l
```

```
++ echo 60/3.28084
```

```
+ a=18.28799941478401872691
```

```
++ bc -l
```

```
++ echo 40/3.28084
```

```
+ b=12.19199960985601248460
```

```
+ echo '18.28799941478401872691 meters x 12.19199960985601248460 meters '
```

```
18.28799941478401872691 meters x 12.19199960985601248460 meters
```

c. Calculate area of 25 such plots in acres.

Solution :

```
$ cat plots.sh
```

```
#!/bin/bash -x
```

```
f=43560
```

```
a=$((60 * 40))
```

```
b=$((a * 25))
```

```
c=$(echo $b / $f | bc -l)
```

```
echo "Area of 25 such plots in acres = "$c
```

```
./plots.sh
```

```
+ f=43560
```

```
+ a=2400
```

```

+ b=60000
++ bc -l
++ echo 60000 / 43560
+ c=1.37741046831955922865
+ echo 'Area of 25 such plots in acres = 1.37741046831955922865'
Area of 25 such plots in acres = 1.37741046831955922865

```

SELECTION PRACTICE PROBLEMS WITH IF & ELSE

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

Solution :

```

$ cat Q1.sh
#!/bin/bash -x
a=$(( RANDOM %1000 ))
b=$(( RANDOM %1000 ))
c=$(( RANDOM %1000 ))
d=$(( RANDOM %1000 ))
e=$(( RANDOM %1000 ))
echo "5 Random Numbers = " $a $b $c $d $e
num=" $a $b $c $d $e "
max=0
for n in $num
do
    if [ $n -gt $max ]
    then
        max=$n
    else
        max=$max
    fi
done
min=`echo $num | awk '{print $1}'`
for m in $num
do
    if [ $m -lt $min ]
    then
        min=$m
    else
        min=$min
    fi
done

```

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

\$./Q1.sh

```
+ a=181
+ b=518
+ c=811
+ d=540
+ e=468
+ echo '5 Random Numbers = ' 181 518 811 540 468
5 Random Numbers = 181 518 811 540 468
+ num='181 518 811 540 468 '
+ max=0
+ for n in $num
+ '[' 181 -gt 0 ']'
+ max=181
+ for n in $num
+ '[' 518 -gt 181 ']'
+ max=518
+ for n in $num
+ '[' 811 -gt 518 ']'
+ max=811
+ for n in $num
+ '[' 540 -gt 811 ']'
+ max=811
+ for n in $num
+ '[' 468 -gt 811 ']'
+ max=811
++ echo 181 518 811 540 468
++ awk '{print $1}'
+ min=181
+ for m in $num
+ '[' 181 -lt 181 ']'
+ min=181
+ for m in $num
+ '[' 518 -lt 181 ']'
+ min=181
+ for m in $num
+ '[' 811 -lt 181 ']'
+ min=181
+ for m in $num
```

```

+ '[' 540 -lt 181 ']'
+ min=181
+ for m in $num
+ '[' 468 -lt 181 ']'
+ min=181
+ echo 'The maximum number is 811'
The maximum number is 811
+ echo 'The minimum number is 181'
The minimum number is 181

```

Ques 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.

Solution :

```

$ cat Q2.sh
#!/bin/bash -x
read -p "Enter Date : " a
b=$(( RANDOM %31 + 1 ))
read -p "Enter Month in number(1 to 12) : " month
day=$a
year=2020
if [[ $month -eq 2 && $day -gt 29 ]]
then
    echo " February consists of 28 days only and 29 days in leap year. Please enter the
correct date."
else
    echo "Date:$day/$month/$year"
    if [[ $month -ge 3 && $month -le 6 ]]
    then
        if [[ $month -eq 3 && $day -ge 20 ]]
        then
            echo "TRUE"
        elif [[ $month -eq 6 && $day -le 20 ]]
        then
            echo "TRUE"
        elif [[ $month -gt 3 && $month -lt 6 ]]
        then
            echo "TRUE"
        else
            echo "FALSE"
        fi
    else

```

```

        echo "FALSE"
    fi
fi

$ ./Q2.sh
+ read -p 'Enter Date : ' a
Enter Date : 23
+ b=13
+ read -p 'Enter Month in number(1 to 12) : ' month
Enter Month in number(1 to 12) : 5
+ day=23
+ year=2020
+ [[ 5 -eq 2 ]]
+ echo Date:23/5/2020
Date:23/5/2020
+ [[ 5 -ge 3 ]]
+ [[ 5 -le 6 ]]
+ [[ 5 -eq 3 ]]
+ [[ 5 -eq 6 ]]
+ [[ 5 -gt 3 ]]
+ [[ 5 -lt 6 ]]
+ echo TRUE
TRUE

```

Ques 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.

Solution :

```

$ cat Q3.sh
#!/bin/bash -x
read -p "Enter Year : " year
if [ $(( $year % 400 )) -eq 0 ]
then
    echo "$year is Leap Year"
elif [[ $(( $year % 100 )) -eq 0 ]]
then
    echo "$year is not Leap Year"
elif [ $(( $year % 4 )) -eq 0 ]
then
    echo "$year is Leap Year"

```

```
else
    echo "$year is not Leap Year"
fi
```

\$./Q3.sh

```
+ read -p 'Enter Year : ' year
Enter Year : 2020
+ '[' 20 -eq 0 ']'
+ [[ 20 -eq 0 ]]
+ '[' 0 -eq 0 ']'
+ echo '2020 is Leap Year'
2020 is Leap Year
```

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

Solution :

\$ cat Q4.sh

```
#!/bin/bash -x
a=`echo $(( RANDOM %2 ))`
if [ $a == 0 ]
then
    echo "HEAD"
else
    echo "TAIL"
fi
```

\$./Q4.sh

```
++ echo 1
+ a=1
+ '[' 1 == 0 ']'
+ echo TAIL
TAIL
```

SELECTION PRACTICE PROBLEMS WITH IF, ELIF & ELSE

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

Solution :

\$ cat Q1.sh

```
#!/bin/bash -x
read -p "Enter Single Digit Number (0-9): " 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 "Error!!! Enter Single Digit Number..."
fi
```

\$./Q1.sh

```
+ read -p 'Enter Single Digit Number (0-9): ' n
Enter Single Digit Number (0-9): 5
+ '[' 5 -eq 0 ']
```

```
+ '[' 5 -eq 1 ']'
+ '[' 5 -eq 2 ']'
+ '[' 5 -eq 3 ']'
+ '[' 5 -eq 4 ']'
+ '[' 5 -eq 5 ']'
+ echo Five
Five
```

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

Solution :

\$ cat Q2.sh

```
#!/bin/bash -x
read -p "Enter week day number : " 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 the valid week day number from 1 to 7 only"
fi
```

\$./Q2.sh

```
+ read -p 'Enter week day number : ' n
Enter week day number : 3
```

```
+ '[' 3 -eq 1 ']'
+ '[' 3 -eq 2 ']'
+ '[' 3 -eq 3 ']'
+ echo Tuesday
Tuesday
```

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

Solution :

```
$ cat Q3.sh
#!/bin/bash -x
read -p "Enter any number : " n
if [[ $n -ge 1 && $n -lt 10 ]]
then
    echo "Ones"
elif [[ $n -ge 10 && $n -lt 100 ]]
then
    echo "Ten"
elif [[ $n -ge 100 && $n -lt 1000 ]]
then
    echo "Hundred"
elif [[ $n -ge 1000 && $n -lt 10000 ]]
then
    echo "Thousand"
elif [[ $n -ge 10000 && $n -lt 100000 ]]
then
    echo "Ten Thousand"
elif [[ $n -ge 100000 && $n -lt 1000000 ]]
then
    echo "Lakhs"
elif [[ $n -ge 1000000 && $n -lt 10000000 ]]
then
    echo "Million"
else
    echo "Enter the number between 1 to Million"
fi
```

```
$ ./Q3.sh
+ read -p 'Enter any number : ' n
Enter any number : 55555
+ [[ 55555 -ge 1 ]]
+ [[ 55555 -lt 10 ]]
```

```

+ [[ 55555 -ge 10 ]]
+ [[ 55555 -lt 100 ]]
+ [[ 55555 -ge 100 ]]
+ [[ 55555 -lt 1000 ]]
+ [[ 55555 -ge 1000 ]]
+ [[ 55555 -lt 10000 ]]
+ [[ 55555 -ge 10000 ]]
+ [[ 55555 -lt 100000 ]]
+ echo 'Ten Thousand'
Ten Thousand

```

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

1. $a + b \times c$
2. $a \% b + c$
3. $c + a / b$
4. $a \times b + c$

Solution :

```

$ cat Q4.sh
#!/bin/bash -x
read -p "Enter first number : " a
read -p "Enter second number : " b
read -p "Enter third number : " c
w=`echo $(( $a + $b * $c ))`
x=`echo $(( $a % $b + $c ))`
y=`echo $(( $c + $a / $b ))`
z=`echo $(( $a * $b + $c ))`
num=" $w $x $y $z "
max=0
max=`echo $num | awk '{print $1}'`
for n in $num
do
if [ $n -gt $max ]
then
max=$n
else
max=$max
fi
done
min=`echo $num | awk '{print $1}'`

```

```

for m in $num
do
if [ $m -lt $min ]
then
min=$m
else
min=$min
fi
done
echo "The maximum among $w, $x, $y, $z : $max"
echo "The minimum among $w, $x, $y, $z : $min"

```

\$./Q4.sh

```

+ read -p 'Enter first number : ' a
Enter first number : 3
+ read -p 'Enter second number : ' b
Enter second number : 4
+ read -p 'Enter third number : ' c
Enter third number : 5
++ echo 23
+ w=23
++ echo 8
+ x=8
++ echo 5
+ y=5
++ echo 17
+ z=17
+ num=' 23 8 5 17 '
+ max=0
++ echo 23 8 5 17
++ awk '{print $1}'
+ max=23
+ for n in $num
+ '[' 23 -gt 23 ']'
+ max=23
+ for n in $num
+ '[' 8 -gt 23 ']'
+ max=23
+ for n in $num
+ '[' 5 -gt 23 ']'
+ max=23
+ for n in $num
+ '[' 17 -gt 23 ']'

```

```

+ max=23
++ echo 23 8 5 17
++ awk '{print $1}'
+ min=23
+ for m in $num
+ '[' 23 -lt 23 ']'
+ min=23
+ for m in $num
+ '[' 8 -lt 23 ']'
+ min=8
+ for m in $num
+ '[' 5 -lt 8 ']'
+ min=5
+ for m in $num
+ '[' 17 -lt 5 ']'
+ min=5
+ echo 'The maximum among 23, 8, 5, 17 : 23'
The maximum among 23, 8, 5, 17 : 23
+ echo 'The minimum among 23, 8, 5, 17 : 5'
The minimum among 23, 8, 5, 17 : 5

```

SELECTION PRACTICE PROBLEMS WITH CASE STATEMENT

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

Solution :

```

$ cat Q1.sh
#!/bin/bash -x
read -p "Enter Single Digit Number : " n
case $n 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"
    ;;
    *)
        echo "Error!!! Enter Single Digit Only"
    ;;
esac

```

\$./Q1.sh

```

+ read -p 'Enter Single Digit Number : ' n
Enter Single Digit Number :3
+ case $n in
+ echo Three
Three

```

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

Solution :

\$ cat Q2.sh

```

#!/bin/bash -x
read -p "Enter any weekday number(1 to 7) : " 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 " Please enter the number from 1 to 7 only"
esac

```

\$./Q2.sh

```

+ read -p 'Enter any weekday number(1 to 7) : ' n
Enter any weekday number(1 to 7) : 4
+ case $n in
+ echo Wednesday
Wednesday

```

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

Solution :

\$ cat Q3.sh

```

#!/bin/bash -x
read -p "Enter any value less than Million : " n
case $n in
"1")
    echo "Ones"
;;
"10")
    echo "Ten"
;;

```



```

"100")
    echo "Hundred"
;;
"1000")
    echo "Thousand"
;;
"10000")
    echo "Ten Thousand"
;;
"100000")
    echo "Lakh"
;;
"1000000")
    echo "Million"
;;
*)
    echo "Error!!! Enter less than or equal to Million";;
esac

```

\$./Q3.sh

```

+ read -p 'Enter any value less than Million : ' n
Enter any value less than Million : 1000
+ case $n in
+ echo Thousand
Thousand

```

Ques 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**

Solution :

\$ cat Q4.sh

```

#!/bin/bash -x
echo -e "OPTIONS :\n1. Feet to inch\n2. Feet to Meter\n3. Inch to Feet\n4. Meter to Feet"
read -p "Enter the option : " n
case $n in
"1")

```

```

        read -p "Enter the feet : " feet
        i=`awk 'BEGIN{printf("%.2f",'$feet' * 12)}'`
        echo "$i inch"
;;
"2")
        read -p "Enter the feet : " feet
        m=`awk 'BEGIN{printf("%.2f",'$feet' * 0.3048)}'`
        echo "$m meters"
;;
"3")
        read -p "Enter the inch : " inch
        f=`awk 'BEGIN{printf("%.2f",'$inch' / 12)}'`
        echo "$f feets";;
"4")
        read -p "Enter the meters : " meter
        f=`awk 'BEGIN{printf("%.2f",'$meter' / 0.3048)}'`
        echo "$f feets"
;;
*)
        echo "Error!!! Enter valid option from 1 to 4 only";;
esac

```

\$./Q4.sh

```
+ echo -e 'OPTIONS :\n1. Feet to inch\n2. Feet to Meter\n3. Inch to Feet\n4. Meter to Feet'
```

```
OPTIONS :
```

1. Feet to inch
2. Feet to Meter
3. Inch to Feet
4. Meter to Feet

```
+ read -p 'Enter the option : ' n
```

```
Enter the option : 2
```

```
+ case $n in
```

```
+ read -p 'Enter the feet : ' feet
```

```
Enter the feet : 30
```

```
++ awk 'BEGIN{printf("%.2f",30 * 0.3048)}'
```

```
+ m=9.144000
```

```
+ echo '9.144000 meters'
```

```
9.144000 meters
```