FOR LOOP

Ques 1. Write a program that takes command line argument n and prints a table of the powers of 2 that are less than or equal to 2^n.

Solution :

```
$ cat Q1.sh
#!/bin/bash -x
read -p "Enter any number : " n
for (( m=1; m<=$n; m++ ))
do
        q=$(( 2**$m ))
        echo "2 power $m is $q"
done
```

```
$ ./Q1.sh
+ read -p 'Enter any number : ' n
Enter any number : 5
+ (( m=1 ))
+ (( m<=5 ))
+ q=2
+ echo '2 power 1 is 2'
2 power 1 is 2
+ (( m++  ))
+ (( m<=5 ))
+ q=4
+ echo '2 power 2 is 4'
2 power 2 is 4
+ (( m++  ))
+ (( m<=5 ))
+ q=8
+ echo '2 power 3 is 8'
2 power 3 is 8
+ (( m++  ))
+ (( m<=5 ))
+ q=16
+ echo '2 power 4 is 16'
2 power 4 is 16
+ (( m++  ))
+ (( m<=5 ))
+ q=32
+ echo '2 power 5 is 32'
```

```
2 power 5 is 32
+ (( m++  ))
+ (( m<=5 ))
```

Ques 2. Write a program that takes command line argument n and prints the nth harmonic number.

Solution :

```
$ cat Q2.sh
#!/bin/bash -x
read -p "Enter the value of n : " n
read -p "Enter the value of a : " a
read -p "enter the value of d : " d
for (( m=0; m<=$n; m++ ))
do
	q=1
	j=$(( $d * $m ))
	aa=$(( $a + $j ))
	echo -ne "$q/$aa "
done
```

```
$ ./Q2.sh
+ read -p 'Enter the value of n : ' n
Enter the value of n : 2
+ read -p 'Enter the value of a : ' a
Enter the value of a : 3
+ read -p 'enter the value of d : ' d
enter the value of d : 4
+ (( m=0 ))
+ (( m<=2 ))
+ q=1
+ j=0
+ aa=3
+ echo -ne '1/3 '
1/3 + (( m++  ))
+ (( m<=2 ))
+ q=1
+ j=4
+ aa=7
+ echo -ne '1/7 '
1/7 + (( m++  ))
+ (( m<=2 ))
+ q=1
+ j=8
+ aa=11
+ echo -ne '1/11 '
1/11 + (( m++  ))
+ (( m<=2 ))

1/3 1/7 1/11
```

Ques 3. Write a program that takes a input and determines if the number is prime.
Solution :

```
$ cat Q3.sh
#!/bin/bash -x
read -p "Enter any number : " n
primenum=1
for (( m=2; m<=$n-1; m++ ))
do
        q=$(( $n % $m ))
        if [ $q -eq '0' ]
        then
                primenum=0
        else
                continue
        fi
done
if [ $primenum -eq 0 ]
then
        echo "$n is not a Prime Number"
else
        echo "$n is Prime Number"
fi

$ ./Q3.sh
+ read -p 'Enter any number : ' n
Enter any number : 3
+ primenum=1
+ (( m=2 ))
+ (( m<=3-1 ))
+ q=1
+ '[' 1 -eq 0 ']'
+ continue
+ (( m++  ))
+ (( m<=3-1 ))
+ '[' 1 -eq 0 ']'
+ echo '3 is Prime Number'
3 is Prime Number
```

Ques 4. Extend the program to take a range of number as input and output the prime numbers in that range.
Solution :

```
$ cat Q4.sh
#!/bin/bash
read -p "Enter the first value : " a
read -p "Enter the last value : " b
primenum=1
for (( j=$a; j<=$b; j++ ))
do
	c=$(( $j - ($j - 2) ))
	primenum=1
	for (( m=$c; m<=$j-1; m++ ))
	do
		q=$(( $j % $m ))
		if [ $q -eq '0' ]
		then
			primenum=0
		else
			continue
		fi
	done
	if [ $primenum -ne 0 ]
	then
		echo -ne $j,
	else
		continue
	fi
done

$ ./Q4.sh
Enter the first value : 2
Enter the last value : 9
2,3,5,7
```

Ques 5. Write a program that computes factorial of a number taken as input.
Solution :

```
$ cat Q5.sh
#!/bin/bash -x
read -p "Enter any number : " n
q=1
for (( m=$n; m>0; m-- ))
do
        q=$(( $q * $m ))
done
echo "Factorial : "$q

$ ./Q5.sh
+ read -p 'Enter any number : ' n
Enter any number : 3
+ q=1
+ (( m=3 ))
+ (( m>0 ))
+ q=3
+ (( m--  ))
+ (( m>0 ))
+ q=6
+ (( m--  ))
+ (( m>0 ))
+ q=6
+ (( m--  ))
+ (( m>0 ))
+ echo 'Factorial : 6'
Factorial : 6
```

Ques 6. Write a program to compute factors of a number N using prime factorization method.
Solution :

```
$ cat Q6.sh
#!/bin/bash -x
read -p "Enter any number : " n
for (( m=2; m<$n; m++ ))
do
```

```
        z=$(( $n % $m ))
        if [ $z -eq 0 ]
        then
                b=$m
                prime=1
                for (( j=2; j<$b-1; j++ ))
                do
                        q=$(( $b % $j ))
                        if [ $q -eq 0 ]
                        then
                                prime=0
                        else
                                continue
                        fi
                done
        if [ $prime -eq 0 ]
        then
                continue
        else
                echo -ne "$b "
        fi
        else
                continue
        fi
done
```

```
$ ./Q6.sh
Enter any number : 100
2 5
```

WHILE LOOP

Ques 1. Write a program that takes command argument n and prints a table of the powers of 2 that are less than or equal to 2 ^n till 256 is reached.
Solution :

```
$ cat Q1.sh
#!/bin/bash -x
read -p "Enter any number : " n
m=1
```

```
while (( m<=$n ))
do
    q=$(( 2**$m ))
    if [ $q -le 256 ]
    then
        echo "2 power $m is $q"
        m=$(( $m + 1 ))
        continue
    else
        echo "Value more than 256"
        break
    fi
done
```

```
$ ./Q1.sh
+ read -p 'Enter any number : ' n
Enter any number : 3
+ m=1
+ ((  m<=3  ))
+ q=2
+ '[' 2 -le 256 ']'
+ echo '2 power 1 is 2'
2 power 1 is 2
+ m=2
+ continue
+ ((  m<=3  ))
+ q=4
+ '[' 4 -le 256 ']'
+ echo '2 power 2 is 4'
2 power 2 is 4
+ m=3
+ continue
+ ((  m<=3  ))
+ q=8
+ '[' 8 -le 256 ']'
+ echo '2 power 3 is 8'
2 power 3 is 8
```

Ques 2. 10. Find The magic Number

a. **Ask the user to think of a number n b/w 1 to 100.**
b. **Then check with the user if the number is less n/2 or greater.**

c. **Repeat till the magic number is reached.**

Solution :

$ cat Q2.sh

```
#!/bin/bash -x
read -p "Enter your lucky number between 1 to 100 : " number
valid=true
lb=1
ub=100
while [ valid ]
do
        mid=$(( $lb+$ub ))
        m=$(( $mid/2 ))
        if [ $number -eq $m ]
        then
                echo $m
                break
        elif [ $number -gt $m ]
        then
                lb=$(( $m+1 ))
        else
                ub=$(( $m-1 ))
        fi
done
```

$ ./Q2.sh

```
+ read -p 'Enter your lucky number between 1 to 100 : ' number
Enter your lucky number between 1 to 100 : 35
+ valid=true
+ lb=1
+ ub=100
+ '[' valid ']'
+ mid=101
+ m=50
+ '[' 35 -eq 50 ']'
+ '[' 35 -gt 50 ']'
+ ub=49
+ '[' valid ']'
+ mid=50
+ m=25
+ '[' 35 -eq 25 ']'
+ '[' 35 -gt 25 ']'
+ lb=26
+ '[' valid ']'
+ mid=75
+ m=37
+ '[' 35 -eq 37 ']'
+ '[' 35 -gt 37 ']'
+ ub=36
+ '[' valid ']'
+ mid=62
+ m=31
+ '[' 35 -eq 31 ']'
+ '[' 35 -gt 31 ']'
+ lb=32
+ '[' valid ']'
+ mid=68
+ m=34
+ '[' 35 -eq 34 ']'
+ '[' 35 -gt 34 ']'
+ lb=35
+ '[' valid ']'
+ mid=71
+ m=35
+ '[' 35 -eq 35 ']'
+ echo 35
35
```

Ques 3. Extend the flip coin problem till either head or tail wins 11 times.
Solution :

```
$ cat Q3.sh
#!/bin/bash -x
heads=0
tails=0
while ( true )
do
    a=`echo $(($RANDOM %2))`
    if [ $a -eq 0 ]
    then
        echo "Heads"
        heads=$(( $heads + 1 ))
        if [ $heads -eq 11 ]
        then
```

```
            break
        fi
    else
        echo "Tails"
        tails=$(( $tails + 1 ))
        if [ $tails -eq 11 ]
        then
            break
        fi
    fi
done
echo "Heads = $heads"
echo "Tails = $tails"
```

```
$ ./Q3.sh
Tails
Tails
Heads
Heads
Heads
Tails
Tails
Tails
Tails
Heads
Heads
Heads
Tails
Heads
Tails
Tails
Tails
Heads
Heads
Tails
Heads = 9
Tails = 11
```

Ques 4. Write a program where a gambler starts with Rs 100 and places Rs 1 bet until he/she goes broke i.e. no more money to gamble or reaches the goal of Rs 200. Keeps the track of number of times won and number of bets made.

Solution :

```
$ cat Q4.sh
#!/bin/bash
n=100
wins=0
losses=0
count=0
while [[ $n -gt 0 && $n -le 200 ]]
do
    a=`echo $(($RANDOM %2))`
    count=$(( $count + 1 ))
    if [ $a -eq 0 ]
    then
        wins=$(( $wins + 1 ))
        n=$(( $n + 1 ))
    else
        losses=$(( $losses + 1 ))
        n=$(( $n - 1 ))
    fi
done
echo "$wins Wins"
echo "$losses Losses"
echo "$count Bets"
```

```
$ ./Q4.sh
1042 Wins
941 Losses
1983 Bets
```

FUNCTIONS

Ques 1. Help user find degf or degc based on their conversion selection . Use case statement and ensure that the inputs are within the freezing point (0° C/32° F) and the boiling point of water (100° C / 212° F).

Solution :

```
$ cat Q1.sh
```

```
#!/bin/bash -x
ftoc() {
	b=`awk 'BEGIN{printf("%0.2f", '$1' - '32')}'`
	degf1=`awk 'BEGIN{printf("%0.2f", '$b' / '1.8')}'`
	echo "$degf1 celsius"
}
ctof() {
	a=`awk 'BEGIN{printf("%0.2f", '$1' * '1.8')}'`
	degc1=`awk 'BEGIN{printf("%0.2f", '$a' + '32')}'`
	echo "$degc1 fahrenheit"
}
echo -e "OPTION :\n1. Fahrenheit to Celsius\n2. Celsius to Fahrenheit"
read -p "Enter the option : " val
case $val in
"1")
	read -p "Enter the value in Fahrenheit : " degc
	k=`awk 'BEGIN{printf("%0.0f", '$degc' + '0')}'`
	l=`awk 'BEGIN{printf("%0.2f", '$degc' + '0')}'`
	if [[ $k -ge 32 && $k -le 212 ]]
	then
		ftoc $l
	else
		echo "Enter the value from 32 to 212 only"
	fi
;;
"2")
	read -p "Enter the value in Celsius : " degf
	s=`awk 'BEGIN{printf("%0.0f", '$degf' + '0')}'`
	t=`awk 'BEGIN{printf("%0.2f", '$degf' + '0')}'`
	if [[ $s -ge 0 && $s -le 100 ]]
	then
		ctof $t
	else
		echo "Enter the value from 0 to 100 celsius only"
	fi
;;
*)
	echo "Error!!! Enter valid input."
;;
esac
```

```
$ ./Q1.sh
+ echo -e 'OPTION :\n1. Fahrenheit to Celsius\n2. Celsius to Fahrenheit'
```

```
OPTION :
1. Fahrenheit to Celsius
2. Celsius to Fahrenheit
+ read -p 'Enter the option : ' val
Enter the option : 1
+ case $val in
+ read -p 'Enter the value in Fahrenheit : ' degc
Enter the value in Fahrenheit : 100
++ awk 'BEGIN{printf("%0.0f", 100 + 0)}'
+ k=100
++ awk 'BEGIN{printf("%0.2f", 100 + 0)}'
+ l=100.00
+ [[ 100 -ge 32 ]]
+ [[ 100 -le 212 ]]
+ ftoc 100.00
++ awk 'BEGIN{printf("%0.2f", 100.00 - 32)}'
+ b=68.00
++ awk 'BEGIN{printf("%0.2f", 68.00 / 1.8)}'
+ degf1=37.78
+ echo '37.78 celsius'
37.78 celsius
```

Ques 2. Write a function to check if the two numbers are palindrome.
Solution :

```
$ cat Q2.sh
#!/bin/bash -x
read -p "Enter the Number : " num
palindrome() {
        val=$1
        a=`echo $val | perl -ne 'chomp;print scalar reverse;'`
        if [ $val -eq $a ]
        then
                echo "The number $val is Palindrome"
        else
                echo "The number $val is not a Palindrome"
        fi
}
palindrome $num
```

```
$ ./Q2.sh
+ read -p 'Enter the Number : ' num
```

```
Enter the Number : 101
+ palindrome 101
+ val=101
++ echo 101
++ perl -ne 'chomp;print scalar reverse;'
+ a=101
+ '[' 101 -eq 101 ']'
+ echo 'The number 101 is Palindrome'
The number 101 is Palindrome
```

Ques 3. Take a number from user and check if the number is prime then show that its palindrome is also prime.

Solution :

```
$ cat Q3.sh
#!/bin/bash
read -p "Enter any number : " n
count=0
funprime() {
    a=$(( $1 - ($1 -2) ))
    primenum=1
    for (( m=$a; m<=$1-1; m++ ))
    do
        q=$(( $1 % $m ))
        if [ $q -eq '0' ]
        then
            primenum=0
        else
            continue
        fi
    done
    if [ $primenum -eq 0 ]
    then
        echo "$1 is not a Prime Number"
    else
        count=1
        echo "$1 is Prime Number"
    fi
}
funprime $n
val=`echo $n | perl -ne 'chomp; print scalar reverse;'`
if [[ $count -eq 1 && $n -eq $val ]]
```

```
then
        echo "$val is Palindrome"
        funprime $val
else
        exit
fi
```

```
$ ./Q3.sh
Enter any number : 151
151 is Prime Number
151 is Palindrome
151 is Prime Number
```