

Assignment No.03

Name: - Omprakash Khawshi

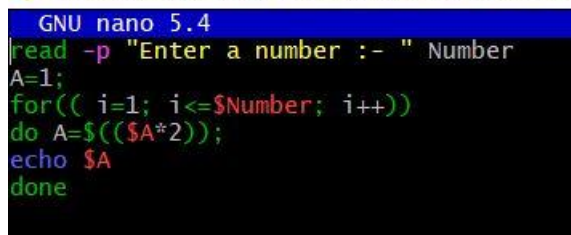
for loop

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

Code: -

```
read -p "Enter a number :- " Number
A=1;
for(( i=1; i<=$Number; i++))
do A=$((A*2));
echo $A
done
```

MINGW64:/d/Assignments/Assignment No.3 for loop



```
GNU nano 5.4
read -p "Enter a number :- " Number
A=1;
for(( i=1; i<=$Number; i++))
do A=$((A*2));
echo $A
done
```

Output: -

MINGW64:/d/Assignments/Assignment No.3 for loop/for loop/Q.1 Write a program that takes a command-line argument n and prints a table of the

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.1 Write a program that takes
a command-line argument n and prints a table of the
$ nano Q1.sh
```

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.1 Write a program that takes
a command-line argument n and prints a table of the
$ ./Q1.sh
```

```
Enter a number :- 4
2
4
8
16
```

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.1 Write a program that takes
a command-line argument n and prints a table of the
$ ./Q1.sh
```

```
Enter a number :- 2
2
4
```

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.1 Write a program that takes
a command-line argument n and prints a table of the
$ ./Q1.sh
```

```
Enter a number :- 8
2
4
8
16
32
64
128
256
```

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

number. Harmonic Number is of the form

$$H_n = 1/1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$$

Code: -

```
read -p "Enter a Number :- " n
i=1
sum=0
while [ $i -le $n ]
do
sum=`expr $sum + \( 10000 / $i \)`
i=`expr $i + 1`
done
echo "Harmonic Number (H$n) is :- "
i=1
while [ $i -le 5 ]
do
a=`echo $sum | cut -c $i`
echo -e "$a\\c"
if [ $i -eq 1 ]
then
echo -e ".\\c"
fi
i=`expr $i + 1`
done
echo
```

MINGW64:/d/Assignments/Assignment No.3 for loop/for



```
GNU nano 5.4
read -p "Enter a Number :- " n
i=1
sum=0
while [ $i -le $n ]
do
sum=`expr $sum + \( 10000 / $i \)`
i=`expr $i + 1`
done
echo "Harmonic Number (H$n) is :- "
i=1
while [ $i -le 5 ]
do
a=`echo $sum | cut -c $i`
echo -e "$a\\c"
if [ $i -eq 1 ]
then
echo -e ".\\c"
fi
i=`expr $i + 1`
done
echo
```

Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for
a command-line argument n and prints the nth harmonic
$ nano Q2.sh

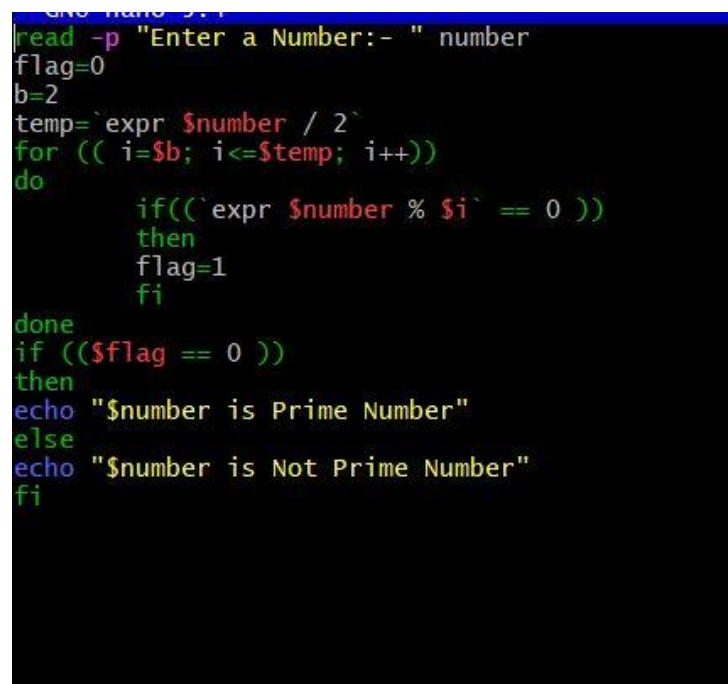
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for
a command-line argument n and prints the nth harmonic
$ ./Q2.sh
Enter a Number :- 5
Harmonic Number (H5) is :-
2.2833

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for
a command-line argument n and prints the nth harmonic
$ ./Q2.sh
Enter a Number :- 9
Harmonic Number (H9) is :-
2.8288
```

Q.3 Write a program that takes a input and determines if the number is a prime.

Code: -

```
read -p "Enter a Number:- " number
flag=0
b=2
temp=`expr $number / 2`
for (( i=$b; i<=$temp; i++))
do
if(`expr $number % $i` == 0 ))
then
flag=1
fi
done
if (($flag == 0 ))
then
echo "$number is Prime Number"
else
echo "$number is Not Prime Number"
fi
```



The screenshot shows a terminal window with a dark background and light-colored text. The script is being executed, and the prompt 'Enter a Number:- ' is visible. The script logic is the same as the one provided in the previous block. The output of the script is not visible in the screenshot.

```
read -p "Enter a Number:- " number
flag=0
b=2
temp=`expr $number / 2`
for (( i=$b; i<=$temp; i++))
do
    if(`expr $number % $i` == 0 ))
    then
        flag=1
    fi
done
if (($flag == 0 ))
then
    echo "$number is Prime Number"
else
    echo "$number is Not Prime Number"
fi
```

Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for lo
$ ./Q3.sh
Enter a Number:- 11
11 is Prime Number

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for lo
$ ./Q3.sh
Enter a Number:- 8
8 is Not Prime Number

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for lo
$ ./Q3.sh
Enter a Number:- 6
6 is Not Prime Number

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for lo
$ ./Q3.sh
Enter a Number:- 2
2 is Prime Number

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for lo
```

Q.4 Extend the program to take a range of number as input and output the Prime Numbers in that range.

Code: -

```
function pn_check()
{
num=$1
flag=0
m=2
temp=`expr $num / 2`
for (( i=$m; i<=$temp; i++))
do
if(`expr $num % $i` == 0 )
then
flag=1
fi
done
if (( $flag == 0 ))
then
echo "$num"
fi
}
read -p "Enter First Number :- " Number_1
read -p "Enter second Number :- " Number_2
echo "Prime numbers betw $Number_1 To $Number_2 :- "
for (( num=$Number_1; num<=$Number_2; num++))
do
pn_check $num
done
```

MINGW64:/d/Assignments/Assignment No.3 for loop/for loop/Q.4 Extend the

```
GNU nano 5.4
function pn_check()
{
num=$1
flag=0
m=2
temp=`expr $num / 2`
for (( i=$m; i<=$temp; i++))
do
if((`expr $num % $i` == 0 ))
then
flag=1
fi
done
if (( $flag == 0 ))
then
echo "$num"
fi
}
read -p "Enter First Number :- " Number_1
read -p "Enter second Number :- " Number_2
echo "Prime numbers betw $Number_1 To $Number_2 :- "
for (( num=$Number_1;num<=$Number_2; num++))
do
pn_check $num
done
```

Output: -

MINGW64:/d/Assignments/Assignment No.3 for loop/for loop/Q.4 Extend the pr

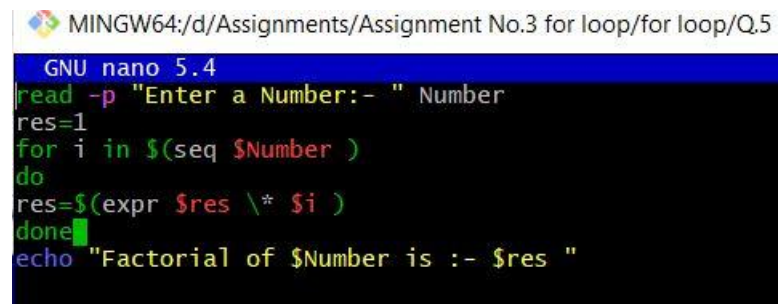
```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for T
a range of number as input and output the Prime Numbers in that
$ ./Q4.sh
Enter First Number :- 1
Enter second Number :- 10
Prime numbers betw 1 To 10 :-
1
2
3
5
7

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for T
a range of number as input and output the Prime Numbers in that
$ ./Q4.sh
Enter First Number :- 11
Enter second Number :- 20
Prime numbers betw 11 To 20 :-
11
13
17
19
```

Q.5 Write a program that computes a factorial of a number taken as input. 5 Factorial – $5! = 1 * 2 * 3 * 4 * 5$

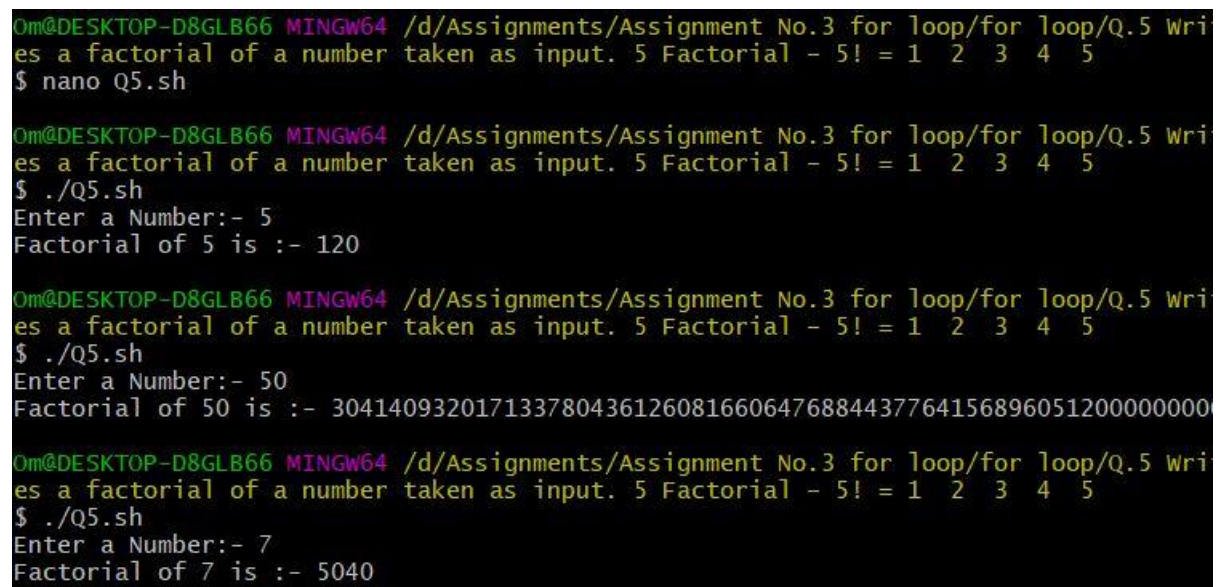
Code: -

```
read -p "Enter a Number:- " Number
res=1
for i in $(seq $Number )
do
res=$(expr $res \* $i )
done
echo "Factorial of $Number is :- $res "
```



```
MINGW64:/d/Assignments/Assignment No.3 for loop/for loop/Q.5
GNU nano 5.4
read -p "Enter a Number:- " Number
res=1
for i in $(seq $Number )
do
res=$(expr $res \* $i )
done
echo "Factorial of $Number is :- $res "
```

Output: -



```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.5 Wri
es a factorial of a number taken as input. 5 Factorial -  $5! = 1 * 2 * 3 * 4 * 5$ 
$ nano Q5.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.5 Wri
es a factorial of a number taken as input. 5 Factorial -  $5! = 1 * 2 * 3 * 4 * 5$ 
$ ./Q5.sh
Enter a Number:- 5
Factorial of 5 is :- 120

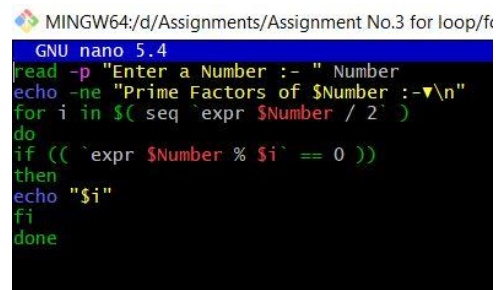
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.5 Wri
es a factorial of a number taken as input. 5 Factorial -  $5! = 1 * 2 * 3 * 4 * 5$ 
$ ./Q5.sh
Enter a Number:- 50
Factorial of 50 is :- 3041409320171337804361260816606476884437764156896051200000000

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.5 Wri
es a factorial of a number taken as input. 5 Factorial -  $5! = 1 * 2 * 3 * 4 * 5$ 
$ ./Q5.sh
Enter a Number:- 7
Factorial of 7 is :- 5040
```


Q.6 Write a program to compute Factors of a number N using prime factorization method. Logic -> Traverse till $i*i \leq N$ instead of $i \leq N$ for efficiency. O/P -> Print the prime factors of number N

Code: -

```
read -p "Enter a Number :- " Number
echo -ne "Prime Factors of $Number :- ▼\n"
for i in $( seq `expr $Number / 2` )
do
if (( `expr $Number % $i` == 0 ))
then
echo "$i"
fi
done
```



```
MINGW64/d/Assignments/Assignment No.3 for loop/fr
GNU nano 5.4
read -p "Enter a Number :- " Number
echo -ne "Prime Factors of $Number :- ▼\n"
for i in $( seq `expr $Number / 2` )
do
if (( `expr $Number % $i` == 0 ))
then
echo "$i"
fi
done
```

Output: -



```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.6 Write a program to compute
Factors of a number N using prime factorization method. Logic - Traverse till ii = N instead of i = N for e
fficiency. OP - Print the prime factors of number N
$ nano Q6.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.6 Write a program to compute
Factors of a number N using prime factorization method. Logic - Traverse till ii = N instead of i = N for e
fficiency. OP - Print the prime factors of number N
$ ./Q6.sh
Enter a Number :- 15
Prime Factors of 15 :- ▼
1
3
5

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.6 Write a program to compute
Factors of a number N using prime factorization method. Logic - Traverse till ii = N instead of i = N for e
fficiency. OP - Print the prime factors of number N
$ nano Q6.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.6 Write a program to compute
Factors of a number N using prime factorization method. Logic - Traverse till ii = N instead of i = N for e
fficiency. OP - Print the prime factors of number N
$ ./Q6.sh
Enter a Number :- 15
Prime Factors of 15 :- ▼
1
3
5

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.6 Write a program to compute
Factors of a number N using prime factorization method. Logic - Traverse till ii = N instead of i = N for e
fficiency. OP - Print the prime factors of number N
$ ./Q6.sh
Enter a Number :- 20
Prime Factors of 20 :- ▼
1
2
4
5
10

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/for loop/Q.6 Write a program to compute
Factors of a number N using prime factorization method. Logic - Traverse till ii = N instead of i = N for e
fficiency. OP - Print the prime factors of number N
$ ./Q6.sh
Enter a Number :- 7
Prime Factors of 7 :- ▼
1
```

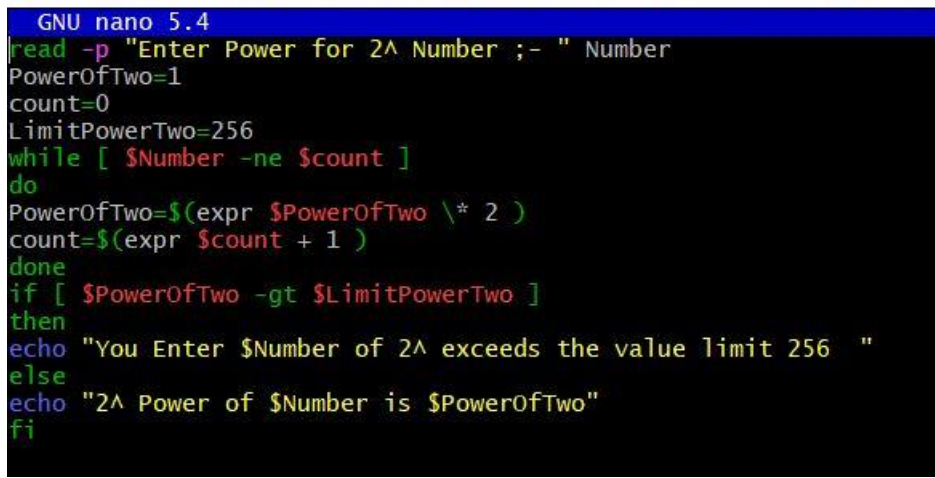
while loop

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

Code: -

```
read -p "Enter Power for 2^ Number ;- " Number
PowerOfTwo=1
count=0
LimitPowerTwo=256
while [ $Number -ne $count ]
do
PowerOfTwo=$(expr $PowerOfTwo \* 2 )
count=$(expr $count + 1 )
done
if [ $PowerOfTwo -gt $LimitPowerTwo ]
then
echo "You Enter $Number of 2^ exceeds the value limit 256 "
else
echo "2^ Power of $Number is $PowerOfTwo"
fi
```

 MINGW64:/d/Assignments/Assignment No.3 for loop/while loop/Q.1 Write a progr



```
GNU nano 5.4
read -p "Enter Power for 2^ Number ;- " Number
PowerOfTwo=1
count=0
LimitPowerTwo=256
while [ $Number -ne $count ]
do
PowerOfTwo=$(expr $PowerOfTwo \* 2 )
count=$(expr $count + 1 )
done
if [ $PowerOfTwo -gt $LimitPowerTwo ]
then
echo "You Enter $Number of 2^ exceeds the value limit 256 "
else
echo "2^ Power of $Number is $PowerOfTwo"
fi
```

Output:-

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.1 Write a program that take
s a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 2
56 is reached
$ nano Q1.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.1 Write a program that take
s a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 2
56 is reached
$ ./Q1.sh
Enter Power for 2^ Number :- 2
2^ Power of 2 is 4

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.1 Write a program that take
s a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 2
56 is reached
$ ./Q1.sh
Enter Power for 2^ Number :- 4
2^ Power of 4 is 16

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.1 Write a program that take
s a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 2
56 is reached
$ ./Q1.sh
Enter Power for 2^ Number :- 12
You Enter 12 of 2^ exceeds the value limit 256

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.1 Write a program that take
s a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 2
56 is reached
$ ./Q1.sh
Enter Power for 2^ Number :- 8
2^ Power of 8 is 256
```

Q.2 Find the Magic Number

- a. Ask the user to think of a number n between 1 to 100
- b. Then check with the user if the number is less than $n/2$ or greater
- c. Repeat till the Magic Number is reached. . . .

Code: -

```
read -p "Think of a Number :- " n
left=1
right=100
while [ $left -le $right ]
do
mid=$(( ($left + $right) / 2 ))
echo "Is Your Number " $mid "?" "True(1) False(0)"
read check
if [ $check -eq 1 ]
then
echo " ☺ ☺ ☺ ☺ Congrats We found your Number"
break
else
echo "Is Your Number smaller than " $mid "True(1) False(0)"
read less
if [ $less -eq 1 ]
then
right=$(( $mid - 1 ))
else
left=$(( $mid + 1 ))
fi
fi
done
```

MINGW64:/d/Assignments/Assignment No.3 for loop/while loop/Q.2 Find the Magic Num1

```
GNU nano 5.4
read -p "Think of a Number :- " n

left=1
right=100
while [ $left -le $right ]
do
    mid=$(( ($left + $right) / 2 ))
    echo "Is Your Number " $mid "?" "True(1) False(0)"
    read check
    if [ $check -eq 1 ]
    then
        echo "==== Congrats We found your Number"
        break
    else
        echo "Is Your Number smaller than " $mid "True(1) False(0)"
        read less
        if [ $less -eq 1 ]
        then
            right=$(( $mid - 1 ))
        else
            left=$(( $mid + 1 ))
        fi
    fi
done
```

Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.2 Find
Ask the user to think of a number n between 1 to 100 b. Then check with the user if th
n2 or greater c. Repeat till the Magic Number is reached
$ nano Q2.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.2 Find
Ask the user to think of a number n between 1 to 100 b. Then check with the user if th
n2 or greater c. Repeat till the Magic Number is reached
$ ./Q2.sh
Think of a Number :- 55
Is Your Number 50 ? True(1) False(0)
0
Is Your Number smaller than 50 True(1) False(0)
0
Is Your Number 75 ? True(1) False(0)
0
Is Your Number smaller than 75 True(1) False(0)
1
Is Your Number 62 ? True(1) False(0)
0
Is Your Number smaller than 62 True(1) False(0)
1
Is Your Number 56 ? True(1) False(0)
0
Is Your Number smaller than 56 True(1) False(0)
1
Is Your Number 53 ? True(1) False(0)
0
Is Your Number smaller than 53 True(1) False(0)
0
Is Your Number 54 ? True(1) False(0)
0
Is Your Number smaller than 54 True(1) False(0)
0
Is Your Number 55 ? True(1) False(0)
1
==== Congrats We found your Number

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.2 Find
then n2 or greater c. Repeat till the Magic Number is reached
```

Code: -

```
RANDOM=$$
Hcount=0
Tcount=0
i=1
b=11
while [ $Hcount -le 11 ] || [ $Tcount -le 11 ]
do
r1=$(expr $RANDOM % 10)
r2=$(expr $r1 % 2)
echo "⊗ Coin Tossed $i Times"
if [ $r2 -eq 0 ]
then
#echo -ne "Head \n"
Hcount=$(expr $Hcount + 1)
elif [ $r2 -eq 1 ]
then
#echo -ne "Tail \n"
Tcount=$(expr $Tcount + 1)
fi
if [ $Hcount -eq 11 ]
then
echo "😊😊😊😊😊😊😊😊😊😊 Head Wins 11 Times 😊😊😊😊😊😊😊😊😊😊"
break
elif [ $Tcount -eq 11 ]
then
echo "😊😊😊😊😊😊😊😊😊😊 Tails wins 11 Times 😊😊😊😊😊😊😊😊😊😊"
break
fi
i=$(expr $i + 1 )
done
```


MINGW64/d/Assignments/Assignment No.3 for loop/while loop/

```
GNU nano 5.4
RANDOM=$$
Hcount=0
Tcount=0
i=1
b=11
while [ $Hcount -le 11 ] | [ $Tcount -le 11 ]
do
r1=$(expr $RANDOM % 10)
r2=$(expr $r1 % 2)
echo "⊗ Coin Tossed $i Times"
if [ $r2 -eq 0 ]
then
#echo -ne "Head \n"
Hcount=$(expr $Hcount + 1)
elif [ $r2 -eq 1 ]
then
#echo -ne "Tail \n"
Tcount=$(expr $Tcount + 1)
fi
if [ $Hcount -eq 11 ]
then
echo "⊗⊗⊗⊗⊗⊗ Head Wins 11 Times ⊗⊗⊗⊗⊗⊗"
break
elif [ $Tcount -eq 11 ]
then
echo "⊗⊗⊗⊗⊗⊗ Tails wins 11 Times ⊗⊗⊗⊗⊗⊗"
break
fi
i=$(expr $i + 1 )
done
```

Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 ~/Downloads
$ ./Q3.sh
⊗ Coin Tossed 1 Times
⊗ Coin Tossed 2 Times
⊗ Coin Tossed 3 Times
⊗ Coin Tossed 4 Times
⊗ Coin Tossed 5 Times
⊗ Coin Tossed 6 Times
⊗ Coin Tossed 7 Times
⊗ Coin Tossed 8 Times
⊗ Coin Tossed 9 Times
⊗ Coin Tossed 10 Times
⊗ Coin Tossed 11 Times
⊗ Coin Tossed 12 Times
⊗ Coin Tossed 13 Times
⊗ Coin Tossed 14 Times
⊗ Coin Tossed 15 Times
⊗ Coin Tossed 16 Times
⊗ Coin Tossed 17 Times
⊗ Coin Tossed 18 Times
⊗ Coin Tossed 19 Times
⊗ Coin Tossed 20 Times
⊗ Coin Tossed 21 Times
⊗⊗⊗⊗⊗⊗ Tails wins 11 Times ⊗⊗⊗⊗⊗⊗

Om@DESKTOP-D8GLB66 MINGW64 ~/Downloads
$ ./Q3.sh
⊗ Coin Tossed 1 Times
⊗ Coin Tossed 2 Times
⊗ Coin Tossed 3 Times
⊗ Coin Tossed 4 Times
⊗ Coin Tossed 5 Times
⊗ Coin Tossed 6 Times
⊗ Coin Tossed 7 Times
⊗ Coin Tossed 8 Times
⊗ Coin Tossed 9 Times
⊗ Coin Tossed 10 Times
⊗ Coin Tossed 11 Times
⊗ Coin Tossed 12 Times
⊗ Coin Tossed 13 Times
⊗ Coin Tossed 14 Times
⊗ Coin Tossed 15 Times
⊗⊗⊗⊗⊗⊗ Head Wins 11 Times ⊗⊗⊗⊗⊗⊗
```

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

Code: -

```
read -p "Enter Amount to Start Play :- " am
temp=$(( $am ))
if [[ $am -gt 100 && $am -lt 200 ]]
then
Won=0
Loss=0
while(( $am > 0 && $am <=200 ))
do
Random_Num=$(( $RANDOM%2 ))
case $Random_Num in
1)
am=$(( $am + 1 ))
#echo -n "$am"
Won=$(( $Won + 1 ))
;;
0)
am=$(( $am - 1 ))
Loss=$(( $Loss + 1 ))
;;
esac
done
echo "The Glamber Invested his Money $temp and he Won $Won Times"
echo "The Glamber Invested his Money $temp and he Loss $Loss Times"
else
echo " Please Enter Amount Range 100 To 200"
fi
```


MINGW64:/d/Assignments/Assignment No.3 for loop/while loop/Q.4 Write a Program where a g

```
GNU nano 5.4
read -p "Enter Amount to Start Play :- " am
temp=$(( $am ))
if [[ $am -gt 100 && $am -lt 200 ]]
then
Won=0
Loss=0
while(( $am > 0 && $am <=200 ))
do
Random_Num=$(( $RANDOM%2 ))
case $Random_Num in
1)
am=$(( $am + 1 ))
#echo -n "$am"
Won=$(( $Won + 1 ))
;;
0)
am=$(( $am - 1 ))
Loss=$(( $Loss + 1 ))
;;
esac
done
echo "The Glamber Invested his Money $temp and he Won $Won Times"
echo "The Glamber Invested his Money $temp and he Loss $Loss Times"
else
echo " Please Enter Amount Range 100 To 200"
fi
```

Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.4 Write a Program where a g
ambler starts with Rs 100 and places Re 1 bet until heshe goes broke i.e. no more money to gamble or reache
s the goal of Rs 200. Keeps track of number of times won and
$ nano Q4.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.4 Write a Program where a g
ambler starts with Rs 100 and places Re 1 bet until heshe goes broke i.e. no more money to gamble or reache
s the goal of Rs 200. Keeps track of number of times won and
$ ./Q4.sh
Enter Amount to Start Play :- 121
The Glamber Invested his Money 121 and he Won 9590 Times
The Glamber Invested his Money 121 and he Loss 9510 Times

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.4 Write a Program where a g
ambler starts with Rs 100 and places Re 1 bet until heshe goes broke i.e. no more money to gamble or reache
s the goal of Rs 200. Keeps track of number of times won and
$ ./Q4.sh
Enter Amount to Start Play :- 110
The Glamber Invested his Money 110 and he Won 5075 Times
The Glamber Invested his Money 110 and he Loss 4984 Times

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.4 Write a Program where a g
ambler starts with Rs 100 and places Re 1 bet until heshe goes broke i.e. no more money to gamble or reache
s the goal of Rs 200. Keeps track of number of times won and
$ ./Q4.sh
Enter Amount to Start Play :- 150
The Glamber Invested his Money 150 and he Won 352 Times
The Glamber Invested his Money 150 and he Loss 301 Times

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/while loop/Q.4 Write a Program where a g
ambler starts with Rs 100 and places Re 1 bet until heshe goes broke i.e. no more money to gamble or reache
s the goal of Rs 200. Keeps track of number of times won and
$ ./Q4.sh
Enter Amount to Start Play :- 201
Please Enter Amount Range 100 To 200
```

Function

Q.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)

a. $\text{degF} = (\text{degC} * 9/5) + 32$

b. $\text{degC} = (\text{degF} - 32) * 5/9$

Code: -

```
function degF()
{
temp=$(( $c * 9 / 5 ) + 32 )
echo "Given Temperature in Degree Fahrenheit is $temp Fahrenheit"
}
function degC()
{
temp=$(( $f - 32 ) * 5 / 9 )
echo "Given Temperature in Degree Celsius is $temp Celsius"
}
echo -ne "1.Degree Celsius To Fahrenheit \n2.Degree Fahrenheit To Degree Celsius \nEnter the choice:-"
read choice
case $choice in
1)
echo -ne "Enter Degree Celsius Temperature :- "
read c
degF $c
;;
2)
echo -ne "Enter Degree Fahrenheit Temperature :- "
read f
degC $f
esac
```

MINGW64:/d/Assignments/Assignment No.3 for loop/Functions/Q.1 Help user find degF or degC based on their Conversion Selection.

```
GNU nano 5.4
function degF()
{
temp=$((($c*9/5)+32))
echo "Given Temprature in Degree Farhenite is $temp Farhenite"
}
function degC()
{
temp=$((($f-32)*5/9))
echo "Given Temprature in Degree Celsius is $temp Celsius"
}
echo -ne "1.Degree Celsius To Farhenite \n2.Degree Farhenite To Degree Celcius \nEnter the coice:-"
read choice
case $choice in
1)
echo -ne "Enter Degree Celsius Temprature :- "
read c
degF $c
;;
2)
echo -ne "Enter Degree Farhenite Temprature :- "
read f
degC $f
esac
```

Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.1 Help user find degF
C based on their Conversion Selection. Use Case Statement and ensure that the inputs are within the
g Point ( 0 °C 32 °F ) and the Boiling Point of Water ( 100
$ nano Q1.sh


Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.1 Help user find degF
C based on their Conversion Selection. Use Case Statement and ensure that the inputs are within the
g Point ( 0 °C 32 °F ) and the Boiling Point of Water ( 100
$ ./Q1.sh
1.Degree Celsius To Farhenite
2.Degree Farhenite To Degree Celcius
Enter the coice:-1
Enter Degree Celsius Temprature :- 55
Given Temprature in Degree Farhenite is 131 Farhenite

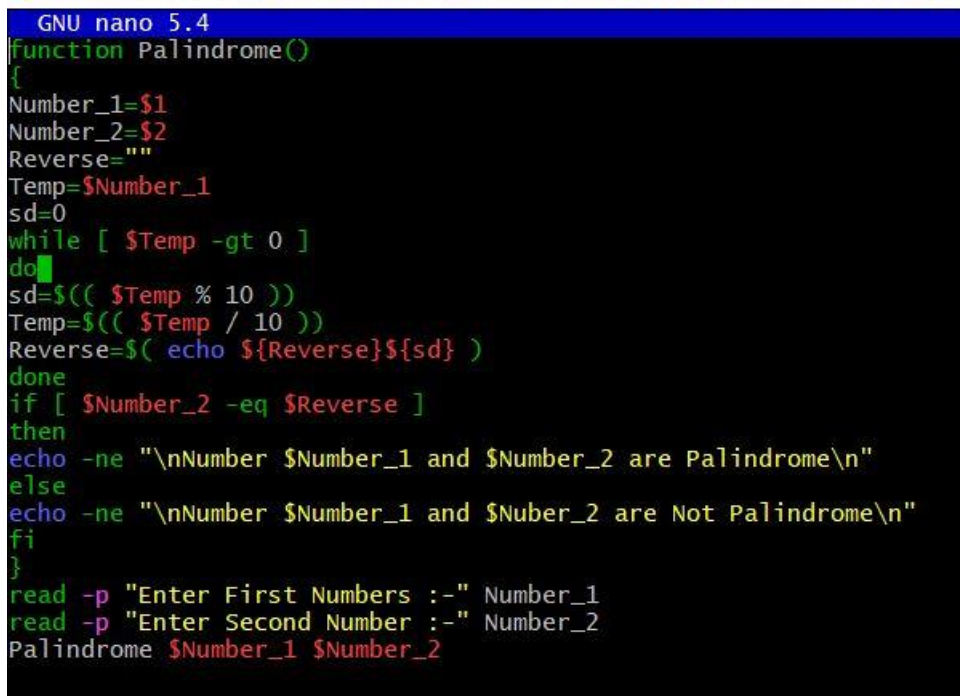
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.1 Help user find degF
C based on their Conversion Selection. Use Case Statement and ensure that the inputs are within the
g Point ( 0 °C 32 °F ) and the Boiling Point of Water ( 100
$ ./Q1.sh
1.Degree Celsius To Farhenite
2.Degree Farhenite To Degree Celcius
Enter the coice:-2
Enter Degree Farhenite Temprature :- 131
Given Temprature in Degree Celsius is 55 Celsius
```

Q.2 Write a function to check if the two numbers are Palindromes

Code: -

```
function Palindrome()
{
Number_1=$1
Number_2=$2
Reverse=""
Temp=$Number_1
sd=0
while [ $Temp -gt 0 ]
do
sd=$(( $Temp % 10 ))
Temp=$(( $Temp / 10 ))
Reverse=$(( echo ${Reverse}${sd} ))
done
if [ $Number_2 -eq $Reverse ]
then
echo -ne "\nNumber $Number_1 and $Number_2 are Palindrome\n"
else
echo -ne "\nNumber $Number_1 and $Nuber_2 are Not Palindrome\n"
fi
}
read -p "Enter First Numbers :-" Number_1
read -p "Enter Second Number :-" Number_2
Palindrome $Number_1 $Number_2
```

 MINGW64:/d/Assignments/Assignment No.3 for loop/Functions/Q.2 Write a function to



```
GNU nano 5.4
function Palindrome()
{
Number_1=$1
Number_2=$2
Reverse=""
Temp=$Number_1
sd=0
while [ $Temp -gt 0 ]
do
sd=$(( $Temp % 10 ))
Temp=$(( $Temp / 10 ))
Reverse=$(( echo ${Reverse}${sd} ))
done
if [ $Number_2 -eq $Reverse ]
then
echo -ne "\nNumber $Number_1 and $Number_2 are Palindrome\n"
else
echo -ne "\nNumber $Number_1 and $Nuber_2 are Not Palindrome\n"
fi
}
read -p "Enter First Numbers :-" Number_1
read -p "Enter Second Number :-" Number_2
Palindrome $Number_1 $Number_2
```

Output: -

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.2 Write a function to check
if the two numbers are Palindromes
$ nano Q2.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.2 Write a function to check
if the two numbers are Palindromes
$ ./Q2.sh
Enter First Numbers :-54
Enter Second Number :-45

Number 54 and 45 are Palindrome

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.2 Write a function to check
if the two numbers are Palindromes
$ ./Q2.sh
Enter First Numbers :-55
Enter Second Number :-58

Number 55 and  are Not Palindrome
```

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

a. Write function check if number is Prime

b. Write function to get the Palindrome.

c. Check if the Palindrome number is also prime

Code: -

```
function Palindrome()
{
Temp=$1
onum=$Temp
sd=0
Reverse=""
while [ $Temp -gt 0 ]
do
sd=$(( $Temp % 10 ))
Temp=$(( $Temp / 10 ))
Reverse=$( echo ${Reverse}${sd} )
done
if [ $onum -eq $Reverse ]
then
echo -ne "\nNumber $onum are Palindrome\n"
else
echo -ne "\nNumber $onum are Not Palindrome\n"
fi
}
function PrimeNumber()
{
Temp=$1
i=2
f=0
while [ $i -le `expr $Temp / 2` ]
do
if [ $(expr $Temp % $i) -eq 0 ]
then
f=1
fi
i=$(expr $i + 1 )
done
if [ $f -eq 1 ]
then
echo -ne "Number $Temp is Not Prime Number"
```

```

else
echo -ne "Number $Temp is Prime Number"
fi
}
echo -ne "1.To Check Prime Number \n2.To check Palindrome Number \n3.To check Prime
as well as palondrome \nEnter Your Choice :- "
read choice
case $choice in
1)
read -p "Enter The Number :-" Temp
PrimeNumber $Temp
;;
2)
read -p "Enter The Number :-" Temp
Palindrome $Temp
;;
3)
read -p "Enter The Number :-" Temp
PrimeNumber $Temp
Palindrome $Temp
esac

```

MINGW64/d:/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is al:




```

GNU nano 5.4 Q3.sh
function Palindrome()
{
Temp=$1
onum=$Temp
sd=0
Reverse=""
while [ $Temp -gt 0 ]
do
sd=$(( $Temp % 10 ))
Temp=$(( $Temp / 10 ))
Reverse=$(( echo ${Reverse}${sd} ))
done
if [ $onum -eq $Reverse ]
then
echo -ne "\nNumber $onum are Palindrome\n"
else
echo -ne "\nNumber $onum are Not Palindrome\n"
fi
}
function PrimeNumber()
{
Temp=$1
i=2
f=0
while [ $i -le `expr $Temp / 2` ]
do
if [ $(expr $Temp % $i) -eq 0 ]
then
f=1
fi
i=$((expr $i + 1 ))
done
if [ $f -eq 1 ]
then
echo -ne "Number $Temp is Not Prime Number"
else
echo -ne "Number $Temp is Prime Number"
fi
}
echo -ne "1.To check Prime Number \n2.To check Palindrome Number \n3.To check Prime as well as palondrome \nEnter Your choice :- "
read choice
case $choice in
1)
read -p "Enter The Number :-" Temp
PrimeNumber $Temp
;;
2)
read -p "Enter The Number :-" Temp
Palindrome $Temp
;;
3)
read -p "Enter The Number :-" Temp
PrimeNumber $Temp
Palindrome $Temp
esac

```


Output: -

 MINGW64:/d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime th

```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. write function check if number is Prime b. write function to get the Palindrome. c. che
$ ./Q3.sh
1.To Check Prime Number
2.To Check Palindrome Number
3.To check Prime as well as palondrome
Enter Your Choice :- 1
Enter The Number :-13
Number 13 is Prime Number
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. write function check if number is Prime b. write function to get the Palindrome. c. che
$ ./Q3.sh
1.To Check Prime Number
2.To Check Palindrome Number
3.To check Prime as well as palondrome
Enter Your Choice :- 1
Enter The Number :-48
Number 48 is Not Prime Number
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. write function check if number is Prime b. write function to get the Palindrome. c. che
$ ./Q3.sh
1.To Check Prime Number
2.To Check Palindrome Number
3.To check Prime as well as palondrome
Enter Your Choice :- 2
Enter The Number :-121
Number 121 are Palindrome
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. write function check if number is Prime b. write function to get the Palindrome. c. che
$ ./Q3.sh
1.To Check Prime Number
2.To Check Palindrome Number
3.To check Prime as well as palondrome
Enter Your Choice :- 2
Enter The Number :-12
Number 12 are Not Palindrome
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. write function check if number is Prime b. write function to get the Palindrome. c. che
$ ./Q3.sh
1.To Check Prime Number
2.To Check Palindrome Number
3.To check Prime as well as palondrome
Enter Your Choice :- 3
Enter The Number :-11
Number 11 is Prime Number
Number 11 are Palindrome
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. write function check if number is Prime b. write function to get the Palindrome. c. che
$ ./Q3.sh
1.To Check Prime Number
2.To Check Palindrome Number
3.To check Prime as well as palondrome
Enter Your Choice :- 3
Enter The Number :-17
```



```
Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. Write function check if number is Prime b. Write function to get the Palindrome. c. Check
$ ./Q3.sh
1.To check Prime Number
2.To check Palindrome Number
3.To check Prime as well as palindrome
Enter Your Choice :- 3
Enter The Number :-17
Number 17 is Prime Number
Number 17 are Not Palindrome

Om@DESKTOP-D8GLB66 MINGW64 /d/Assignments/Assignment No.3 for loop/Functions/Q.3 Take a number from user and check if the number is a Prime then show that its palindrome is also prime a. Write function check if number is Prime b. Write function to get the Palindrome. c. Check
$ ./Q3.sh
1.To check Prime Number
2.To check Palindrome Number
3.To check Prime as well as palindrome
Enter Your Choice :- 3
Enter The Number :-33
Number 33 is Not Prime Number
Number 33 are Palindrome
```