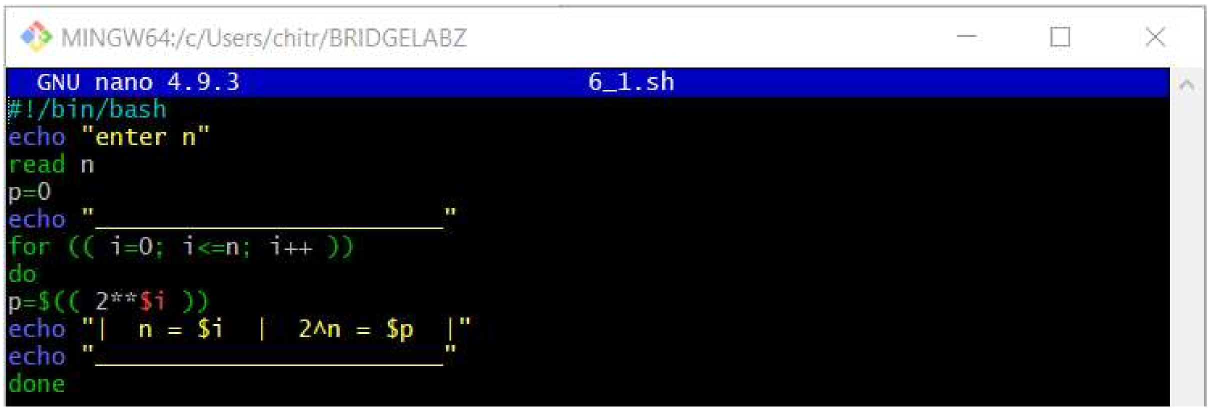
Name: Chitra Shamkalyan Bapat **Date: 15-09-2020**

Day 6 Assignment (3)

FOR LOOP

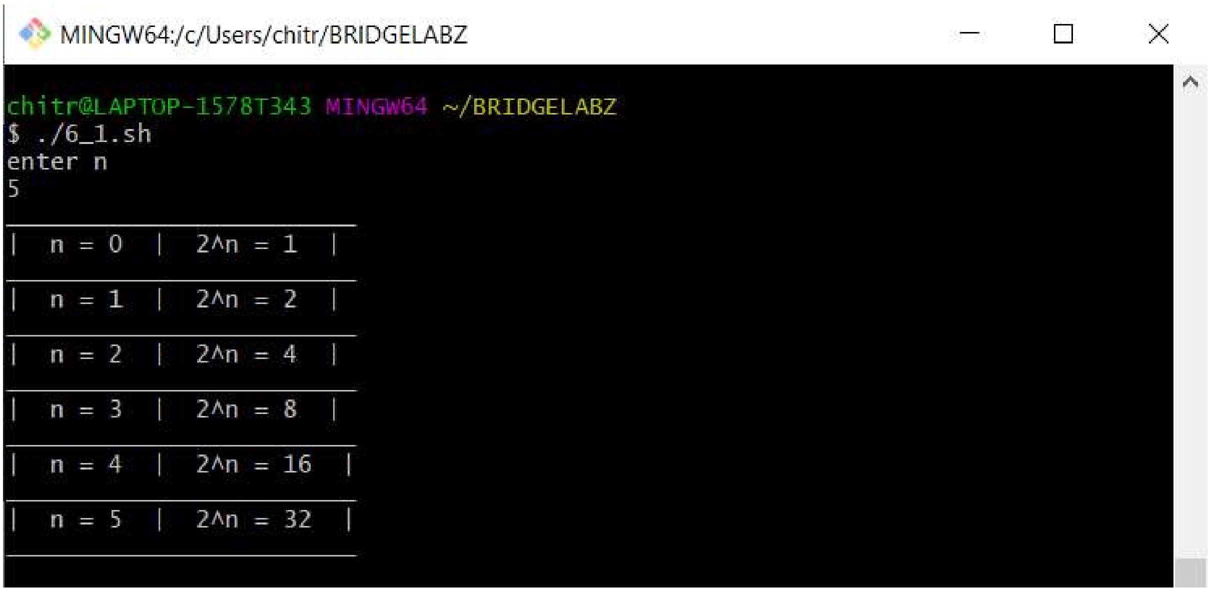
1. Write a program that takes a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n.

PROGRAM:



```
  GNU nano 4.9.3                         6_1.sh
#!/bin/bash
echo "enter n"
read n
p=0
echo "________________________"
for (( i=0; i<=n; i++ ))
do
p=$(( 2**$i ))
echo "|   n = $i   |   2^n = $p   |"
echo "________________________"
done
```

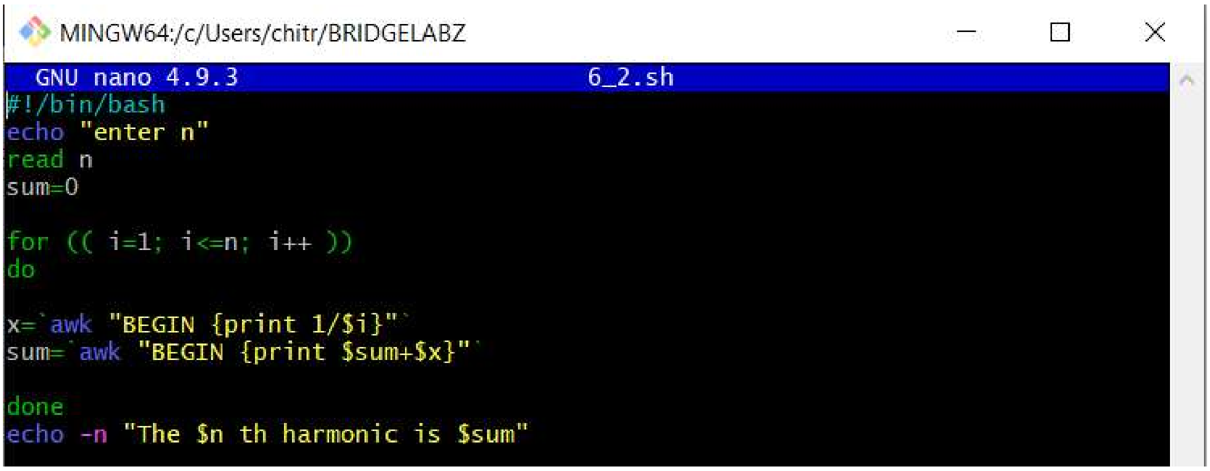
OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_1.sh
enter n
5
________________________
|   n = 0   |   2^n = 1   |
________________________
|   n = 1   |   2^n = 2   |
________________________
|   n = 2   |   2^n = 4   |
________________________
|   n = 3   |   2^n = 8   |
________________________
|   n = 4   |   2^n = 16   |
________________________
|   n = 5   |   2^n = 32   |
________________________
```

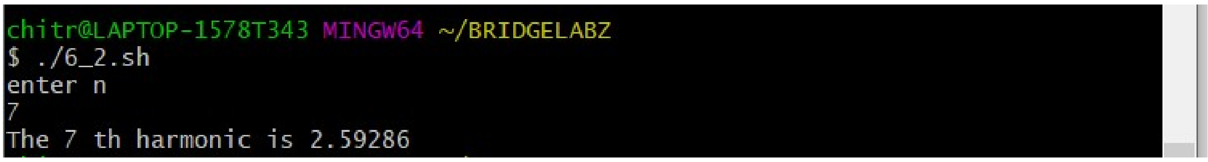
2. Write a program that takes a command-line argument n and prints the nth harmonic number. Harmonic Number is of the form Hn = 1/1 + 1/2 + 1/3 + 1/4 + 1/5 + ... + 1/n

PROGRAM:



```
  GNU nano 4.9.3                         6_2.sh
#!/bin/bash
echo "enter n"
read n
sum=0

for (( i=1; i<=n; i++ ))
do

x=`awk "BEGIN {print 1/$i}"`
sum=`awk "BEGIN {print $sum+$x}"`

done
echo -n "The $n th harmonic is $sum"
```

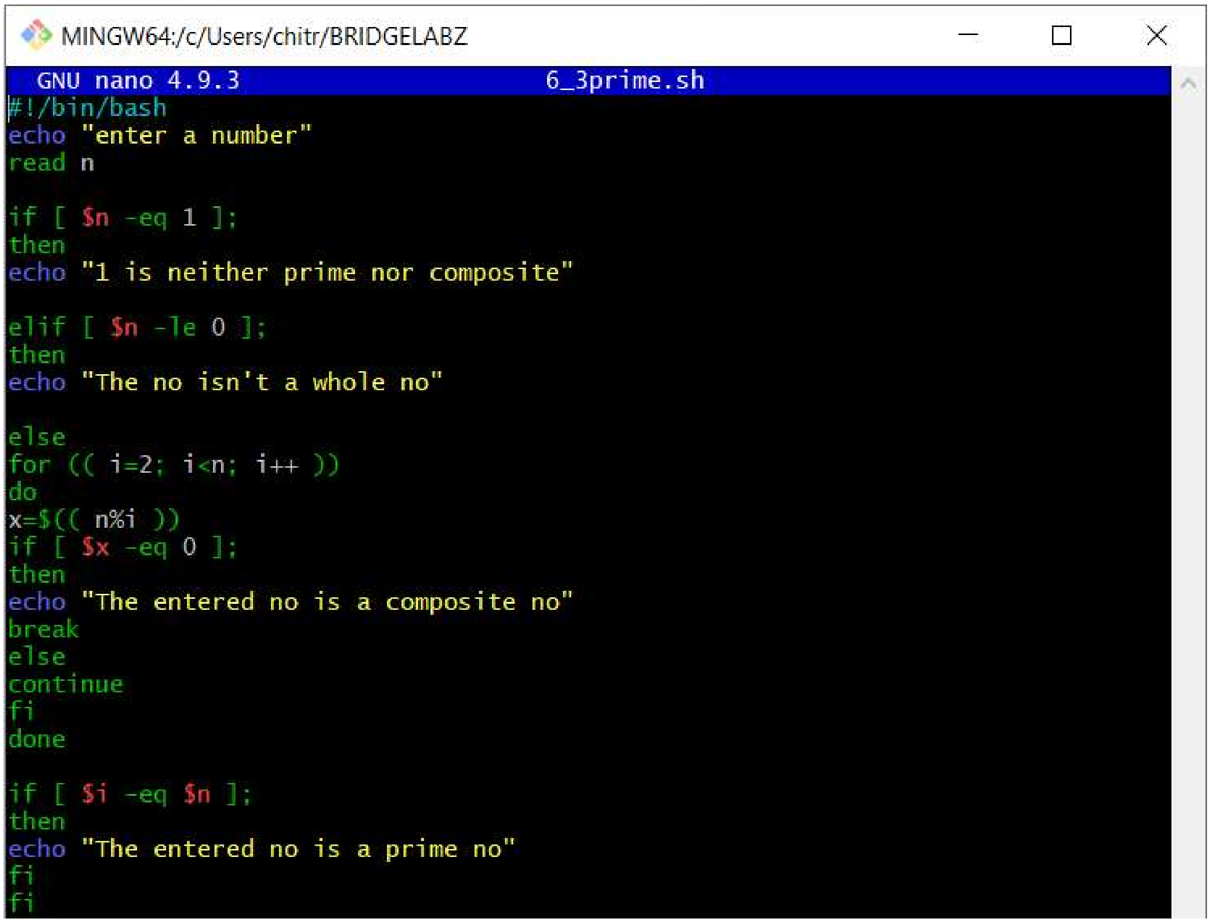
OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_2.sh
enter n
7
The 7 th harmonic is 2.59286
```

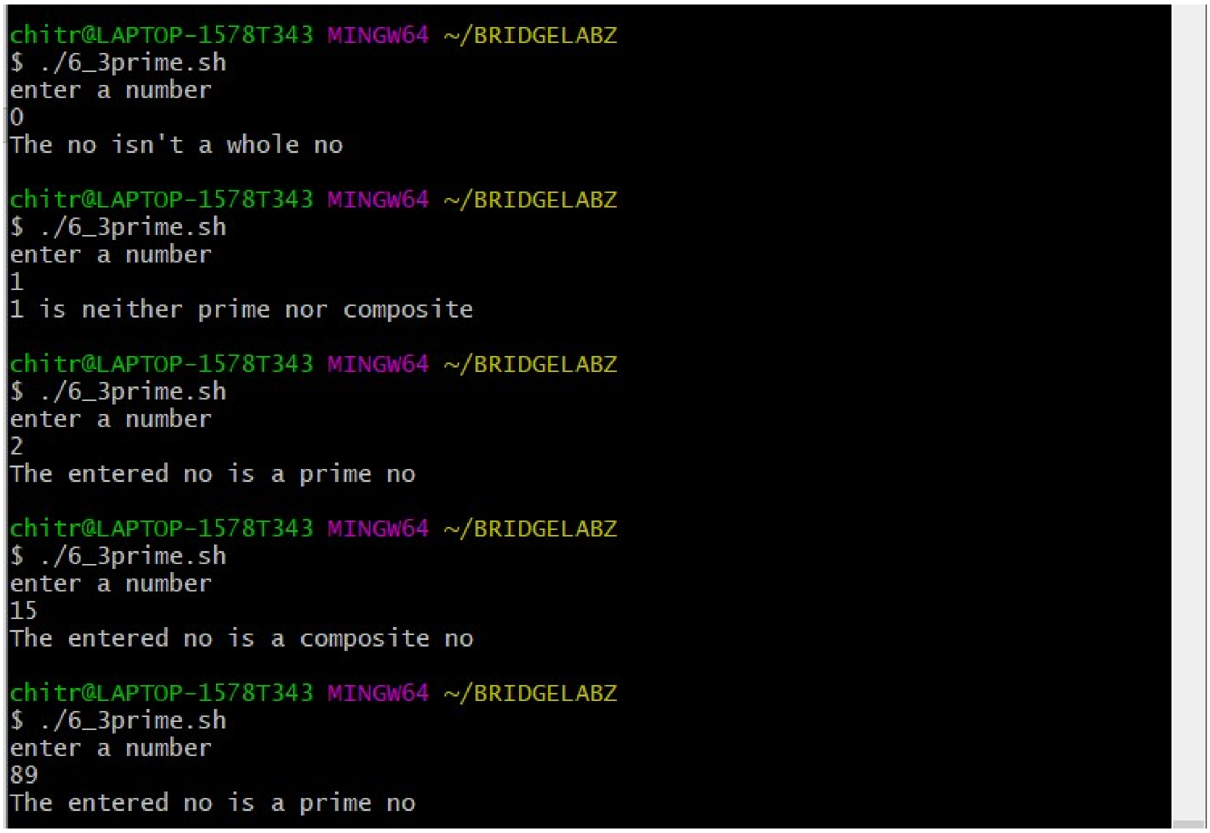
3.Write a program that takes a input and determines if the number is a prime.

PROGRAM:



```
#!/bin/bash
echo "enter a number"
read n

if [ $n -eq 1 ];
then
echo "1 is neither prime nor composite"

elif [ $n -le 0 ];
then
echo "The no isn't a whole no"

else
for (( i=2; i<n; i++ ))
do
x=$(( n%i ))
if [ $x -eq 0 ];
then
echo "The entered no is a composite no"
break
else
continue
fi
done

if [ $i -eq $n ];
then
echo "The entered no is a prime no"
fi
fi
```

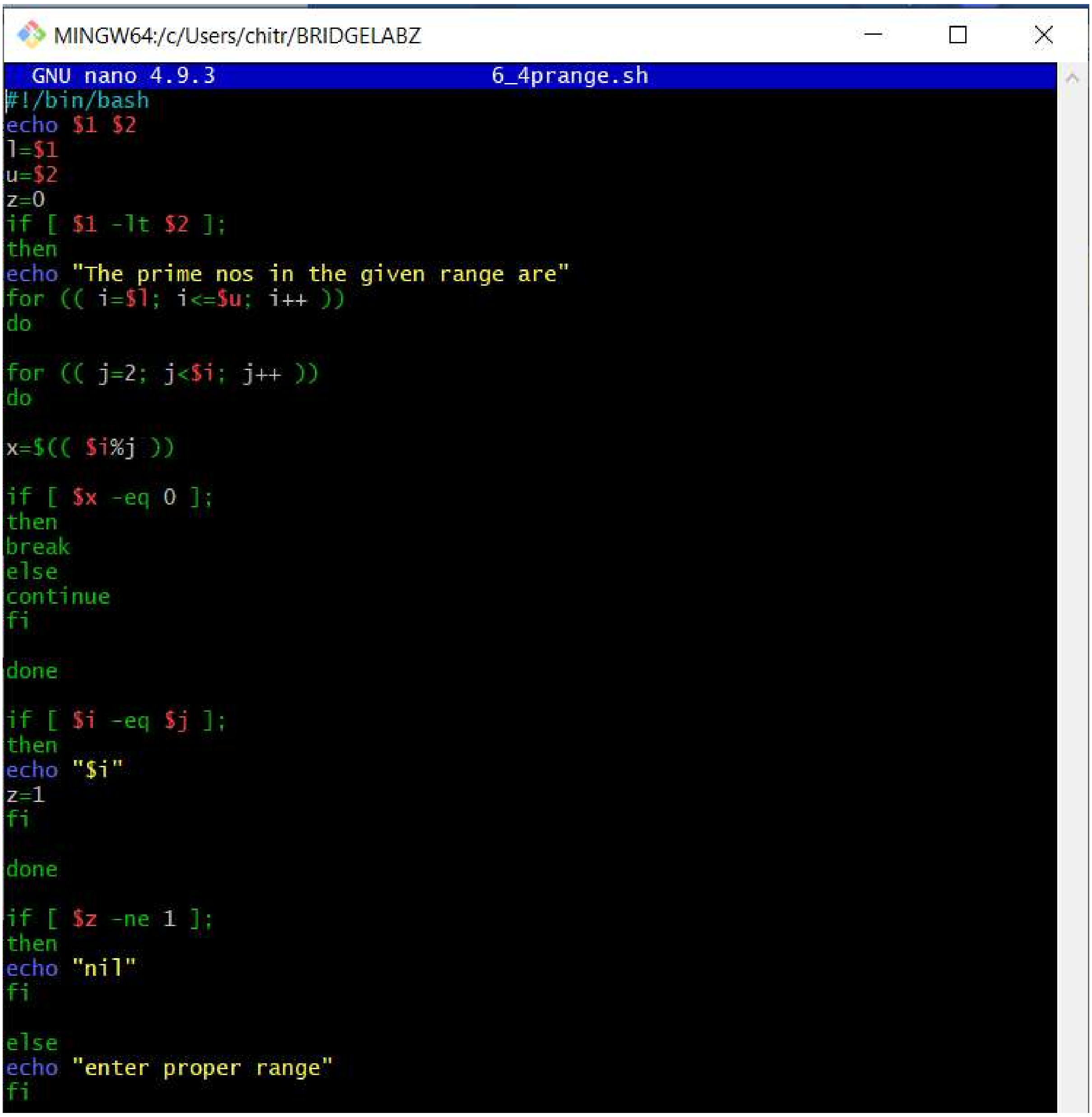
OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_3prime.sh
enter a number
0
The no isn't a whole no

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_3prime.sh
enter a number
1
1 is neither prime nor composite

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_3prime.sh
enter a number
2
The entered no is a prime no

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_3prime.sh
enter a number
15
The entered no is a composite no

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_3prime.sh
enter a number
89
The entered no is a prime no
```

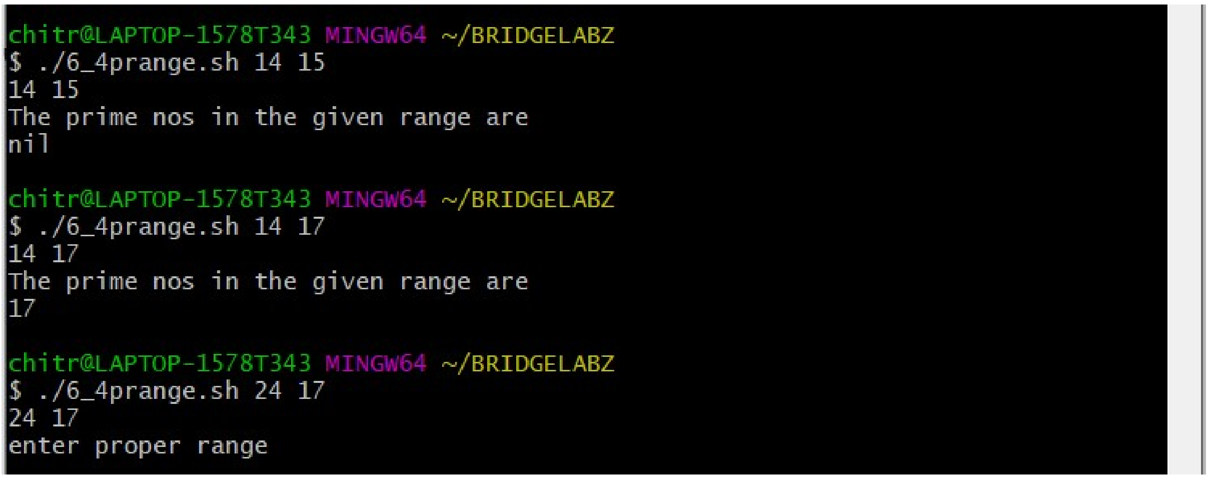
4. Extend the program to take a range of number as input and output the Prime numbers in that range.

PROGRAM:



```
#!/bin/bash
echo $1 $2
l=$1
u=$2
z=0
if [ $1 -lt $2 ];
then
echo "The prime nos in the given range are"
for (( i=$l; i<=$u; i++ ))
do

for (( j=2; j<$i; j++ ))
do

x=$(( $i%j ))

if [ $x -eq 0 ];
then
break
else
continue
fi

done

if [ $i -eq $j ];
then
echo "$i"
z=1
fi

done

if [ $z -ne 1 ];
then
echo "nil"
fi

else
echo "enter proper range"
fi
```

OUTPUT:

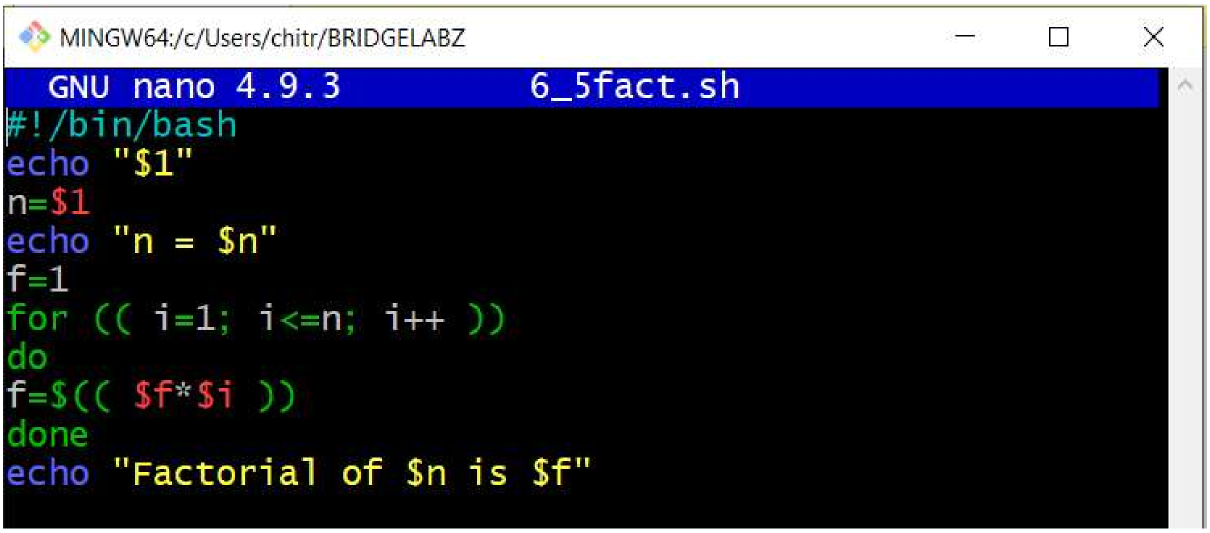


```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_4prange.sh 14 15
14 15
The prime nos in the given range are
nil

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_4prange.sh 14 17
14 17
The prime nos in the given range are
17

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_4prange.sh 24 17
24 17
enter proper range
```

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

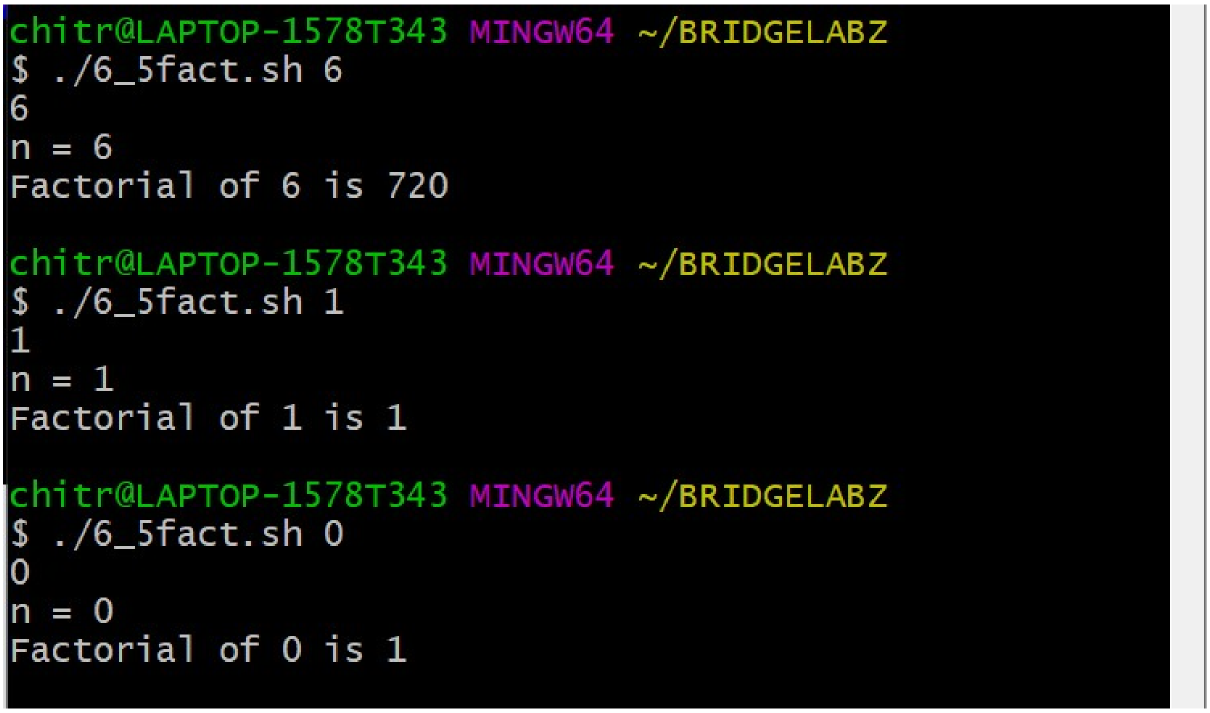
Factorial – 5! = 1 * 2 * 3 * 4 * 5

PROGRAM:



```
  GNU nano 4.9.3          6_5fact.sh
#!/bin/bash
echo "$1"
n=$1
echo "n = $n"
f=1
for (( i=1; i<=n; i++ ))
do
f=$(( $f*$i ))
done
echo "Factorial of $n is $f"
```

OUTPUT:



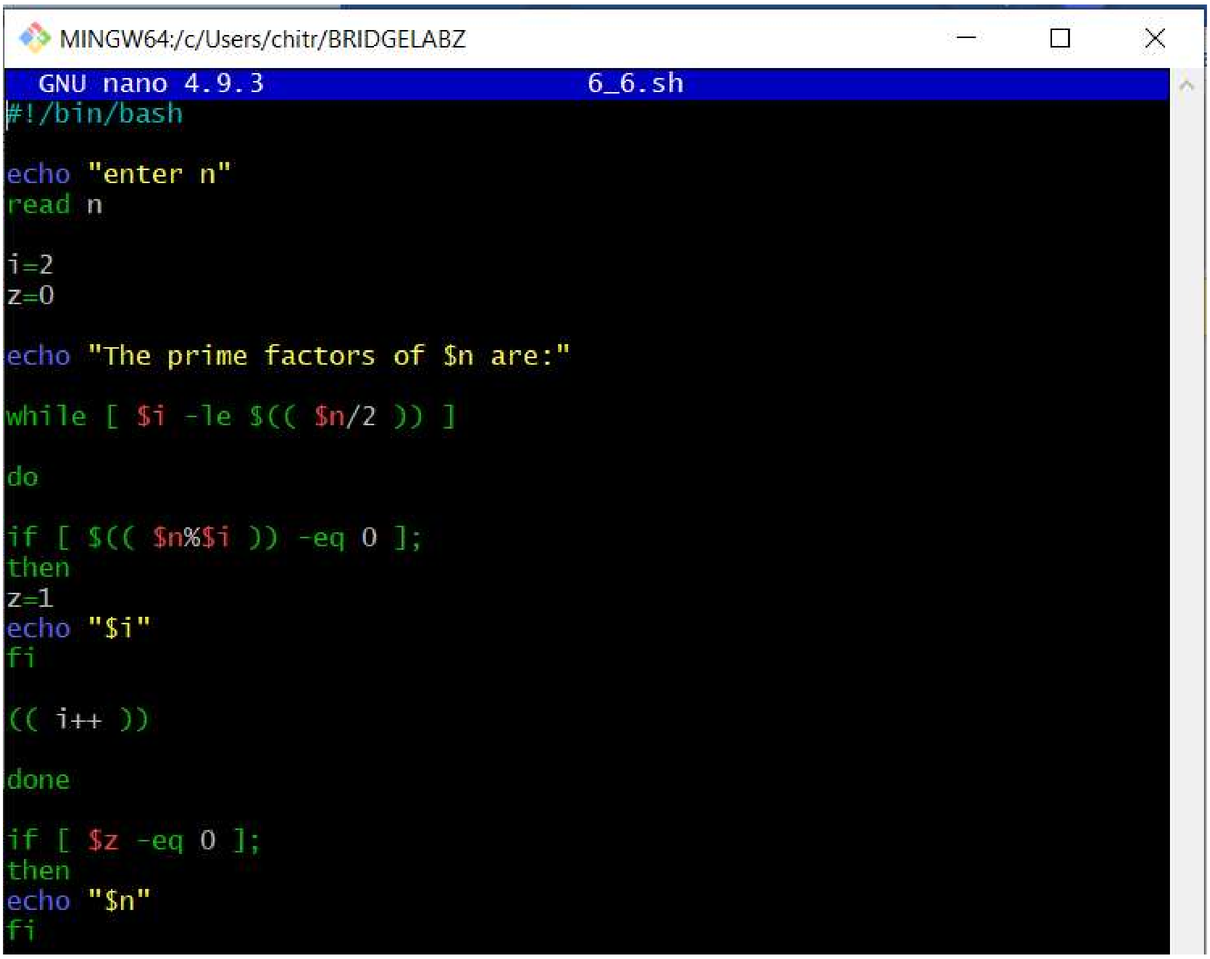
```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_5fact.sh 6
6
n = 6
Factorial of 6 is 720

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_5fact.sh 1
1
n = 1
Factorial of 1 is 1

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_5fact.sh 0
0
n = 0
Factorial of 0 is 1
```

6. Write a program to compute Factors of a number N using prime factorization method.

Logic -> Traverse till i*i <= N instead of i <= N for efficiency.

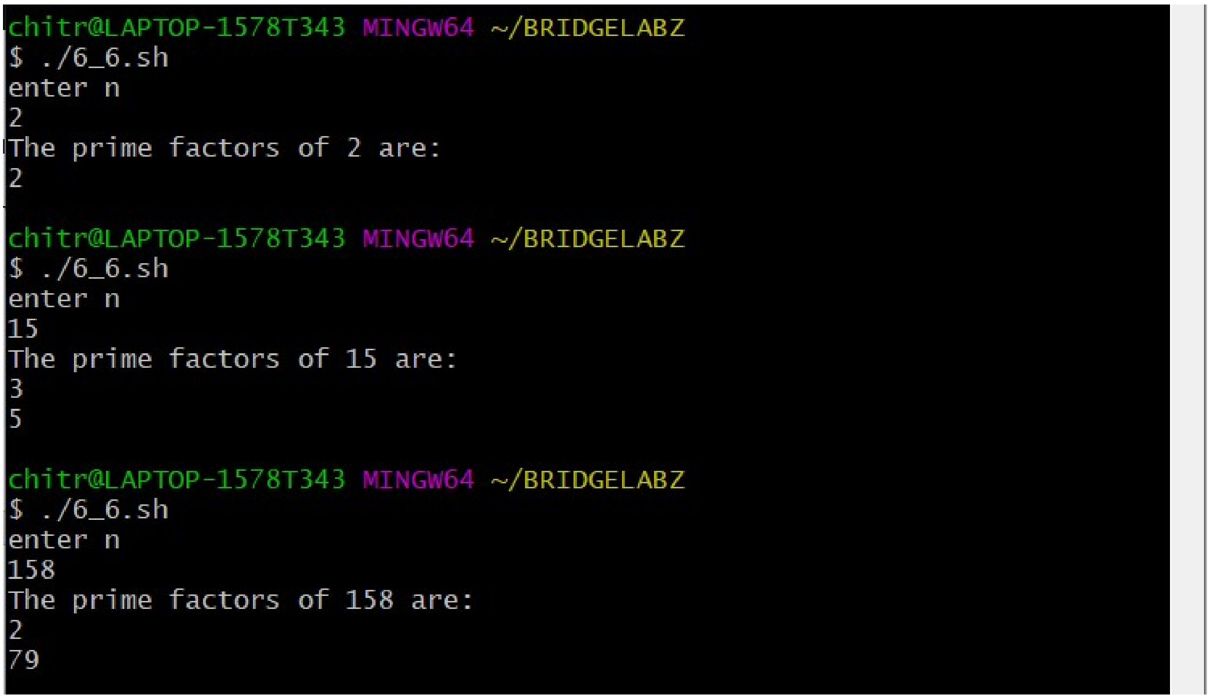
O/P -> Print the prime factors of number N.

PROGRAM:



```
  GNU nano 4.9.3                    6_6.sh
#!/bin/bash

echo "enter n"
read n

i=2
z=0

echo "The prime factors of $n are:"

while [ $i -le $(( $n/2 )) ]

do

if [ $(( $n%$i )) -eq 0 ];
then
z=1
echo "$i"
fi

(( i++ ))

done

if [ $z -eq 0 ];
then
echo "$n"
fi
```

OUTPUT:

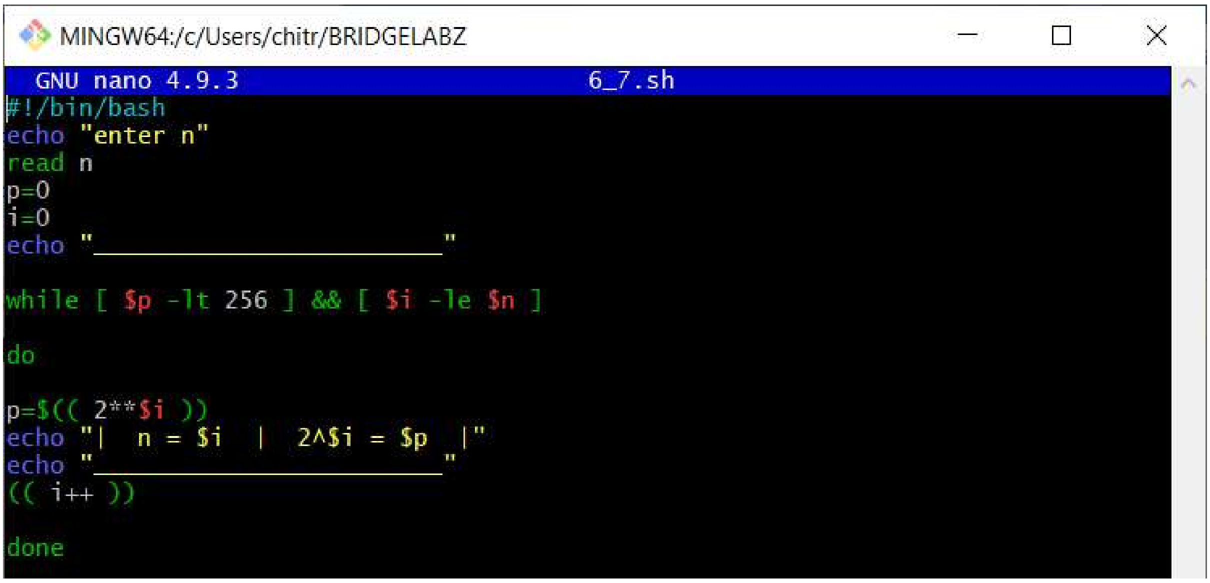


```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_6.sh
enter n
2
The prime factors of 2 are:
2

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_6.sh
enter n
15
The prime factors of 15 are:
3
5

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_6.sh
enter n
158
The prime factors of 158 are:
2
79
```

While Loop

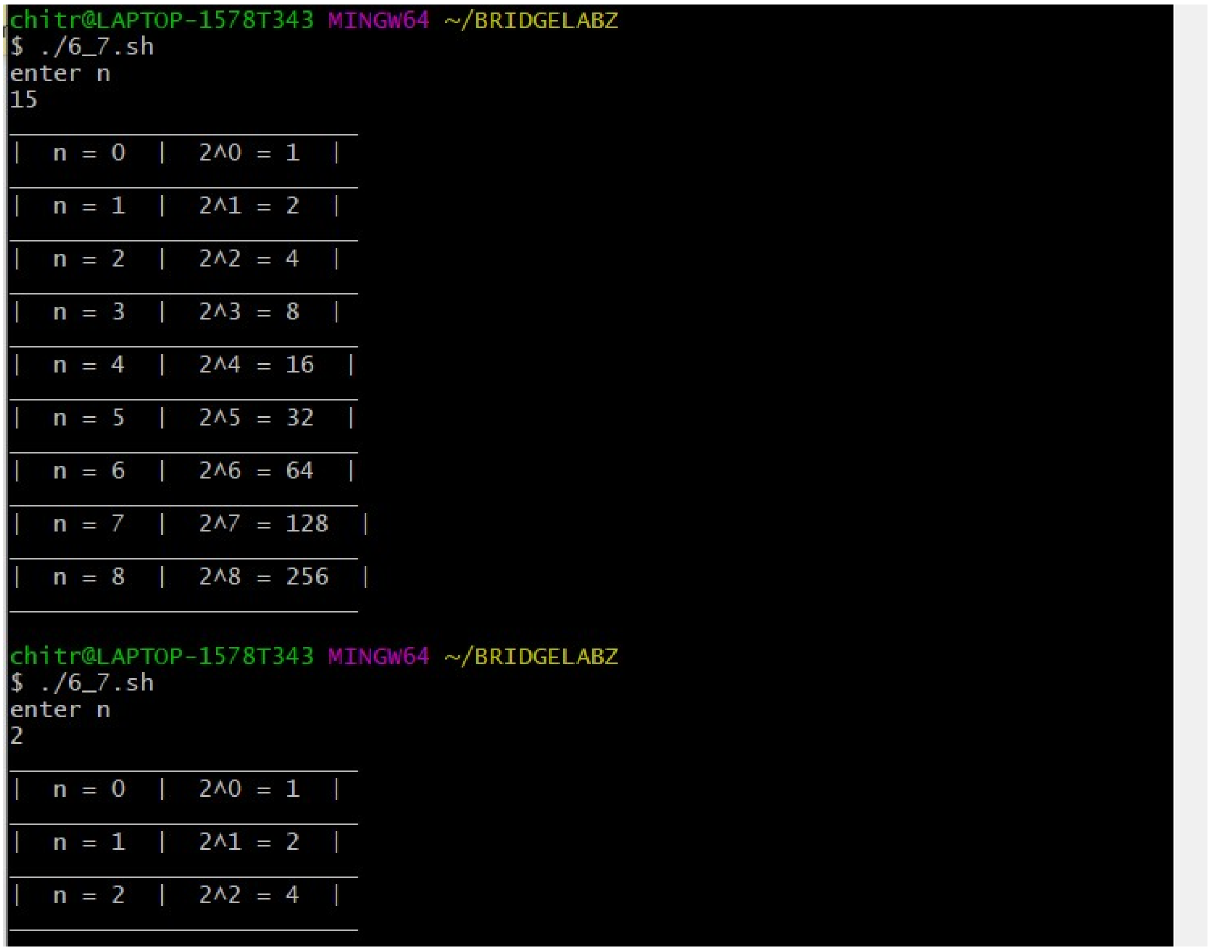
1.Write a program that takes a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 256 is reached.

PROGRAM:



```
  GNU nano 4.9.3                      6_7.sh
#!/bin/bash
echo "enter n"
read n
p=0
i=0
echo "________________________"

while [ $p -lt 256 ] && [ $i -le $n ]

do

p=$(( 2**$i ))
echo "|  n = $i  |  2^$i = $p  |"
echo "________________________"
(( i++ ))

done
```

OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_7.sh
enter n
15
_________________________
|   n = 0   |   2^0 = 1   |
_________________________
|   n = 1   |   2^1 = 2   |
_________________________
|   n = 2   |   2^2 = 4   |
_________________________
|   n = 3   |   2^3 = 8   |
_________________________
|   n = 4   |   2^4 = 16   |
_________________________
|   n = 5   |   2^5 = 32   |
_________________________
|   n = 6   |   2^6 = 64   |
_________________________
|   n = 7   |   2^7 = 128   |
_________________________
|   n = 8   |   2^8 = 256   |
_________________________

chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_7.sh
enter n
2
_________________________
|   n = 0   |   2^0 = 1   |
_________________________
|   n = 1   |   2^1 = 2   |
_________________________
|   n = 2   |   2^2 = 4   |
_________________________
```

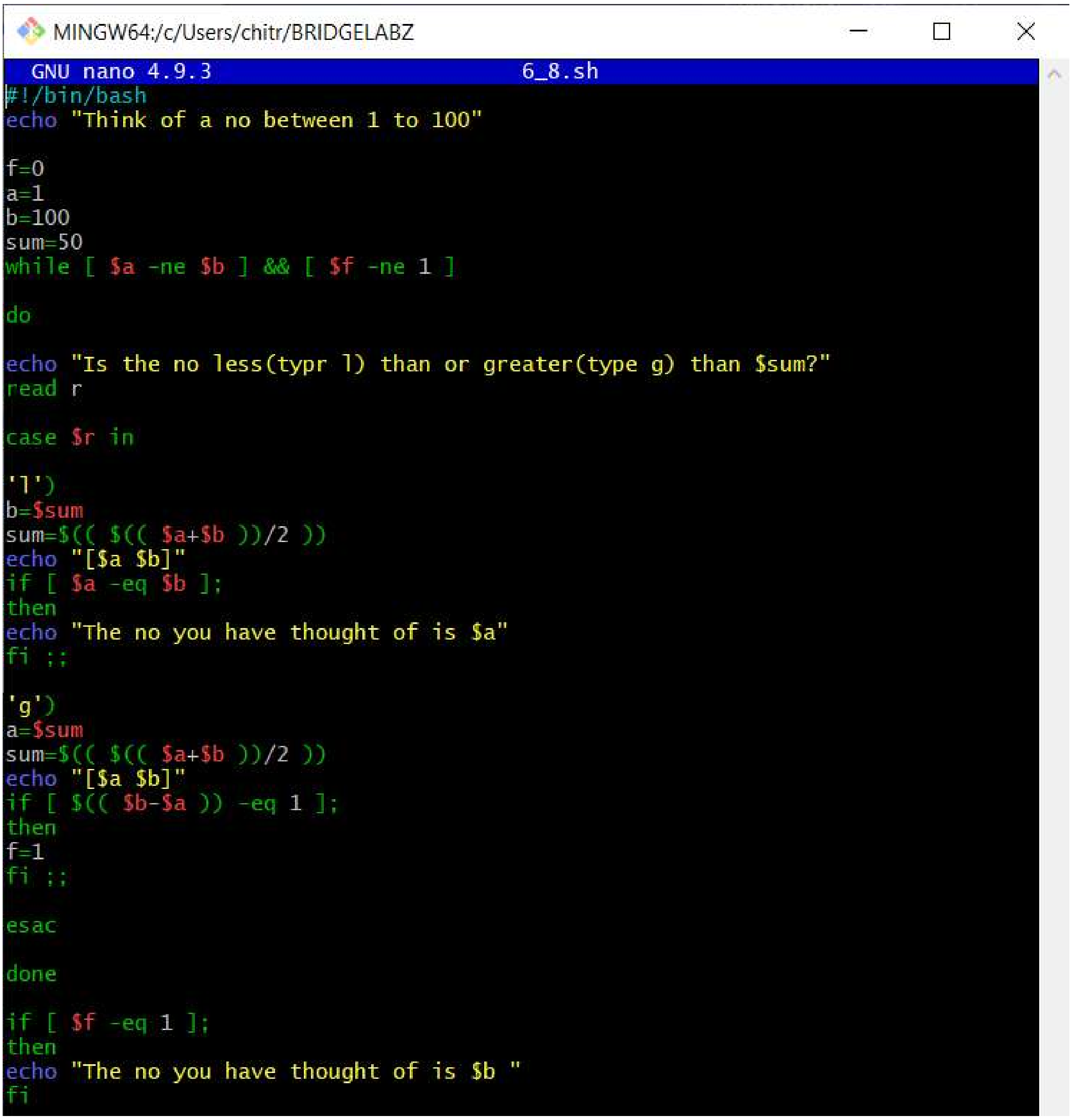
2. Find the Magic Number

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 then n/2 or greater

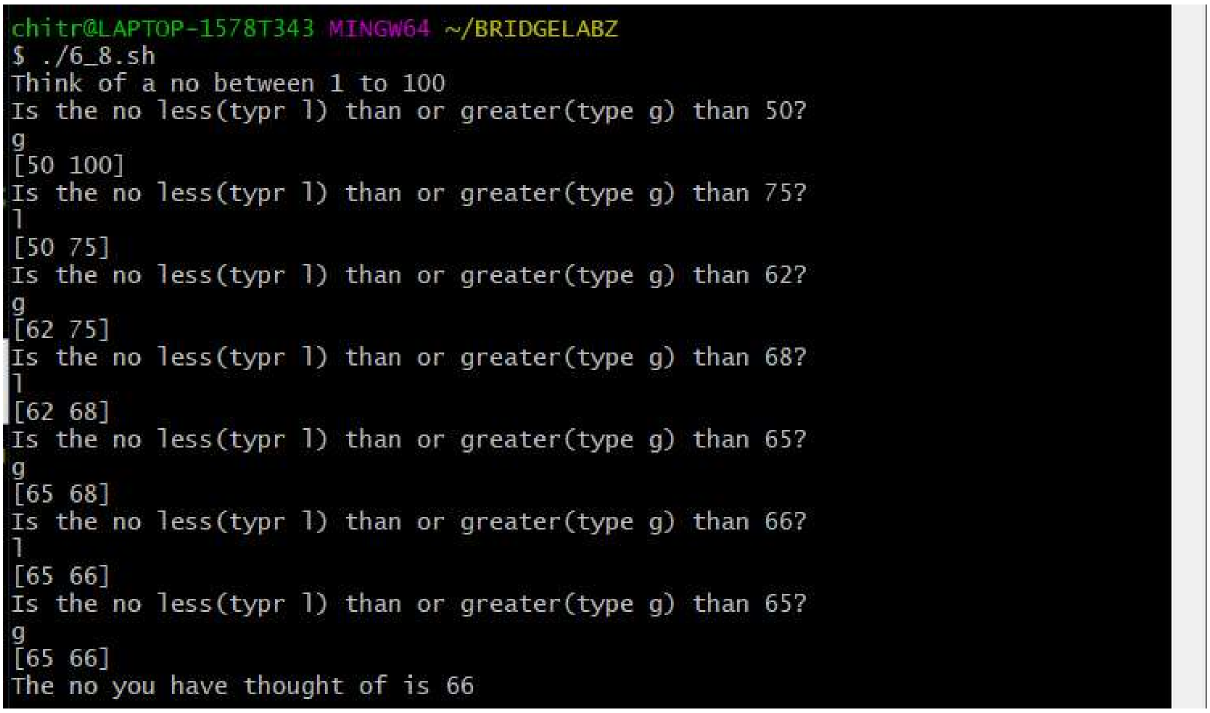
c. Repeat till the Magic Number is reached.

PROGRAM:



```
  GNU nano 4.9.3                      6_8.sh
#!/bin/bash
echo "Think of a no between 1 to 100"

f=0
a=1
b=100
sum=50
while [ $a -ne $b ] && [ $f -ne 1 ]

do

echo "Is the no less(typr l) than or greater(type g) than $sum?"
read r

case $r in

'l')
b=$sum
sum=$(( $(( $a+$b ))/2 ))
echo "[$a $b]"
if [ $a -eq $b ];
then
echo "The no you have thought of is $a"
fi ;;

'g')
a=$sum
sum=$(( $(( $a+$b ))/2 ))
echo "[$a $b]"
if [ $(( $b-$a )) -eq 1 ];
then
f=1
fi ;;

esac

done

if [ $f -eq 1 ];
then
echo "The no you have thought of is $b "
fi
```

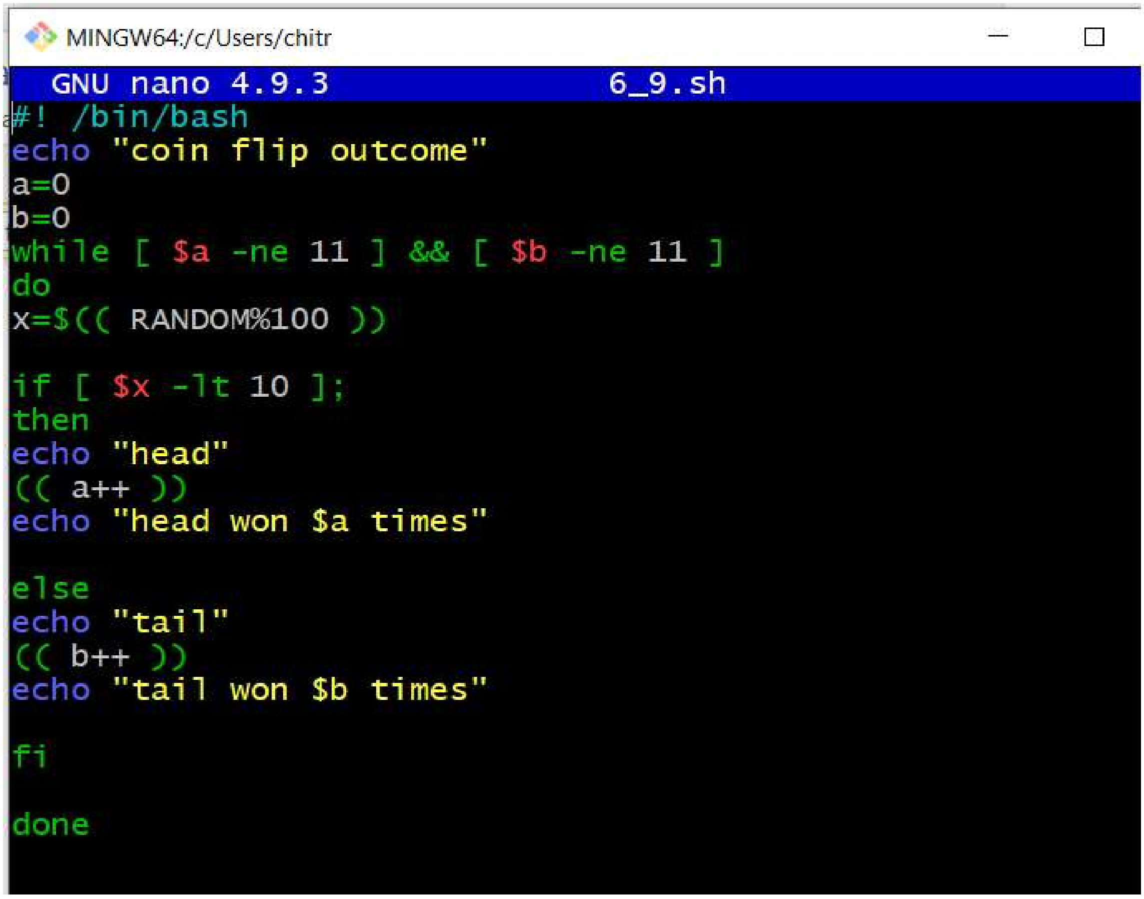
OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_8.sh
Think of a no between 1 to 100
Is the no less(typr l) than or greater(type g) than 50?
g
[50 100]
Is the no less(typr l) than or greater(type g) than 75?
l
[50 75]
Is the no less(typr l) than or greater(type g) than 62?
g
[62 75]
Is the no less(typr l) than or greater(type g) than 68?
l
[62 68]
Is the no less(typr l) than or greater(type g) than 65?
g
[65 68]
Is the no less(typr l) than or greater(type g) than 66?
l
[65 66]
Is the no less(typr l) than or greater(type g) than 65?
g
[65 66]
The no you have thought of is 66
```

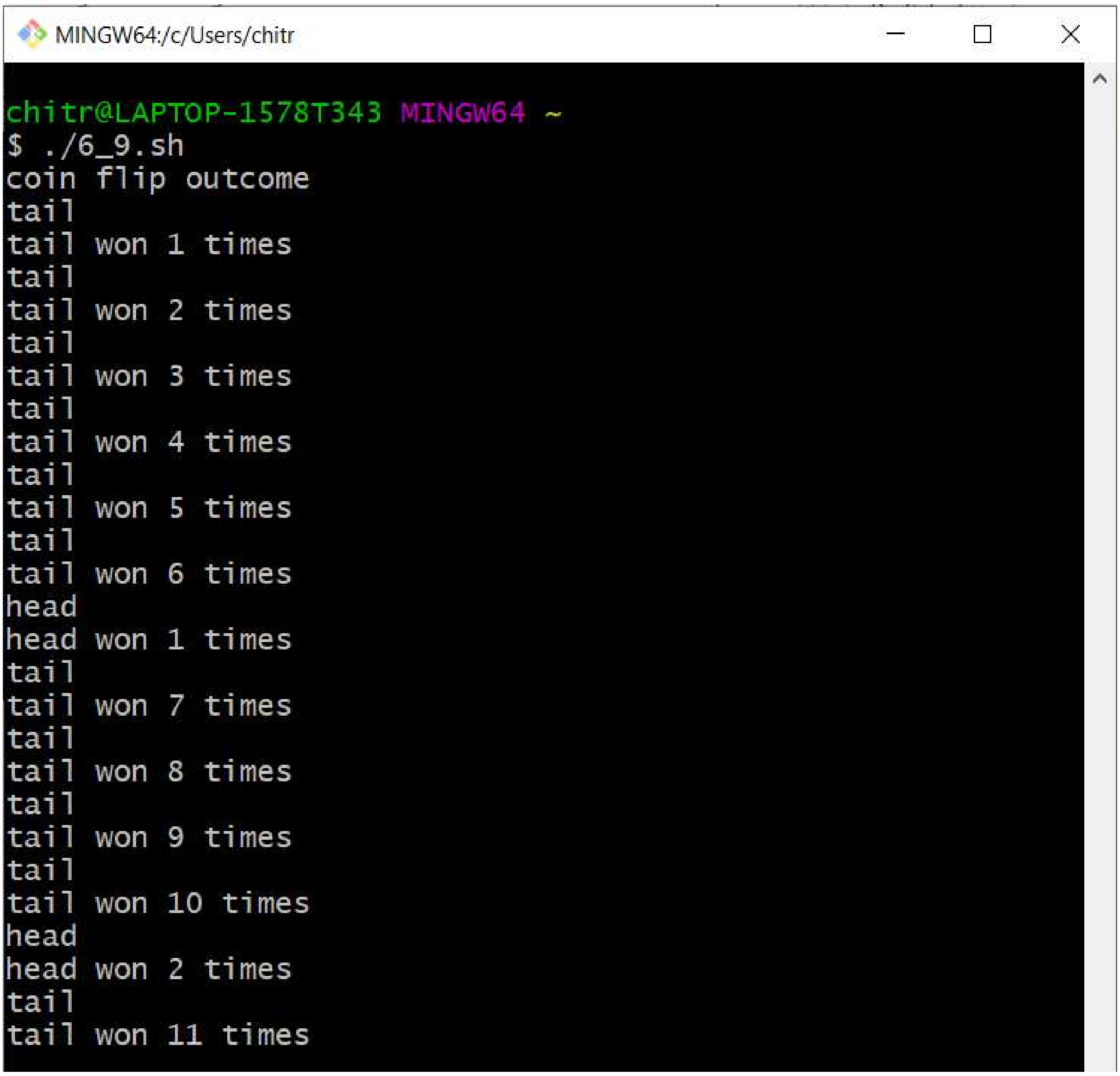
3.Extend the Flip Coin problem till either Heads or Tails wins 11 times.

PROGRAM:



```
  GNU nano 4.9.3                  6_9.sh
#! /bin/bash
echo "coin flip outcome"
a=0
b=0
while [ $a -ne 11 ] && [ $b -ne 11 ]
do
x=$(( RANDOM%100 ))

if [ $x -lt 10 ];
then
echo "head"
(( a++ ))
echo "head won $a times"

else
echo "tail"
(( b++ ))
echo "tail won $b times"

fi

done
```

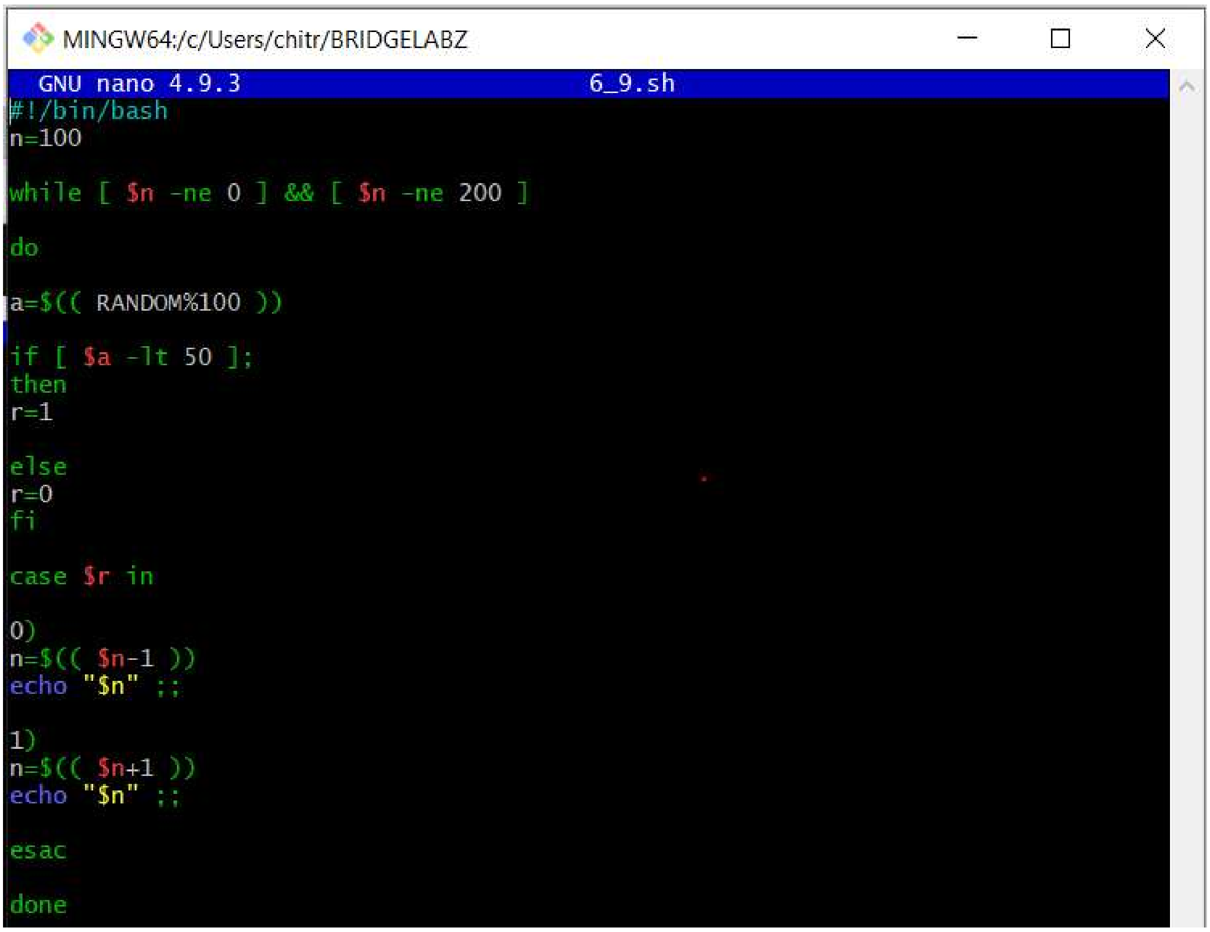
OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_9.sh
coin flip outcome
tail
tail won 1 times
tail
tail won 2 times
tail
tail won 3 times
tail
tail won 4 times
tail
tail won 5 times
tail
tail won 6 times
head
head won 1 times
tail
tail won 7 times
tail
tail won 8 times
tail
tail won 9 times
tail
tail won 10 times
head
head won 2 times
tail
tail won 11 times
```

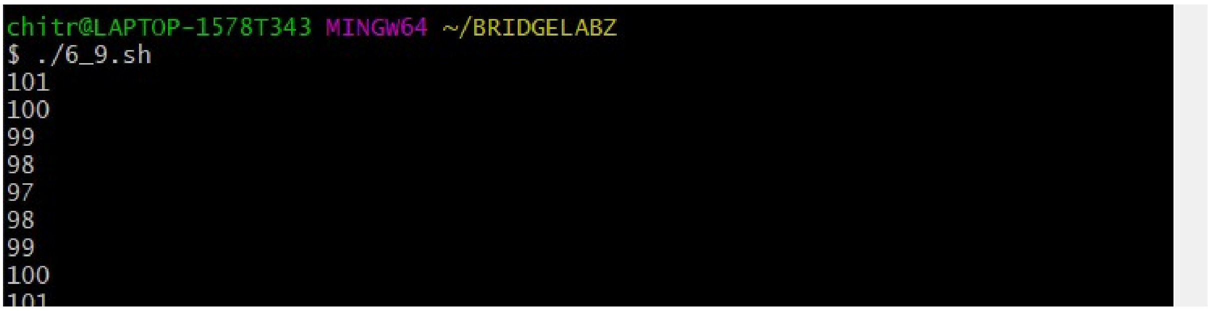
4. Write a Program where a gambler starts with Rs 100 and places Re 1 bet until he/she goes broke i.e. no more money to gamble or reaches the goal of Rs 200. Keeps track of number of times won and number of bets made.

PROGRAM:



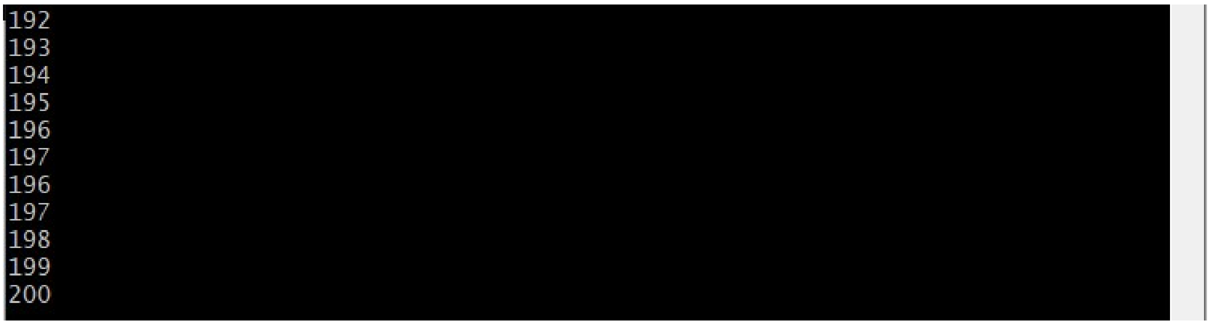
```
#!/bin/bash
n=100

while [ $n -ne 0 ] && [ $n -ne 200 ]

do

a=$(( RANDOM%100 ))

if [ $a -lt 50 ];
then
r=1

else
r=0
fi

case $r in

0)
n=$(( $n-1 ))
echo "$n" ;;

1)
n=$(( $n+1 ))
echo "$n" ;;

esac

done
```

OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~/BRIDGELABZ
$ ./6_9.sh
101
100
99
98
97
98
99
100
101
```

…

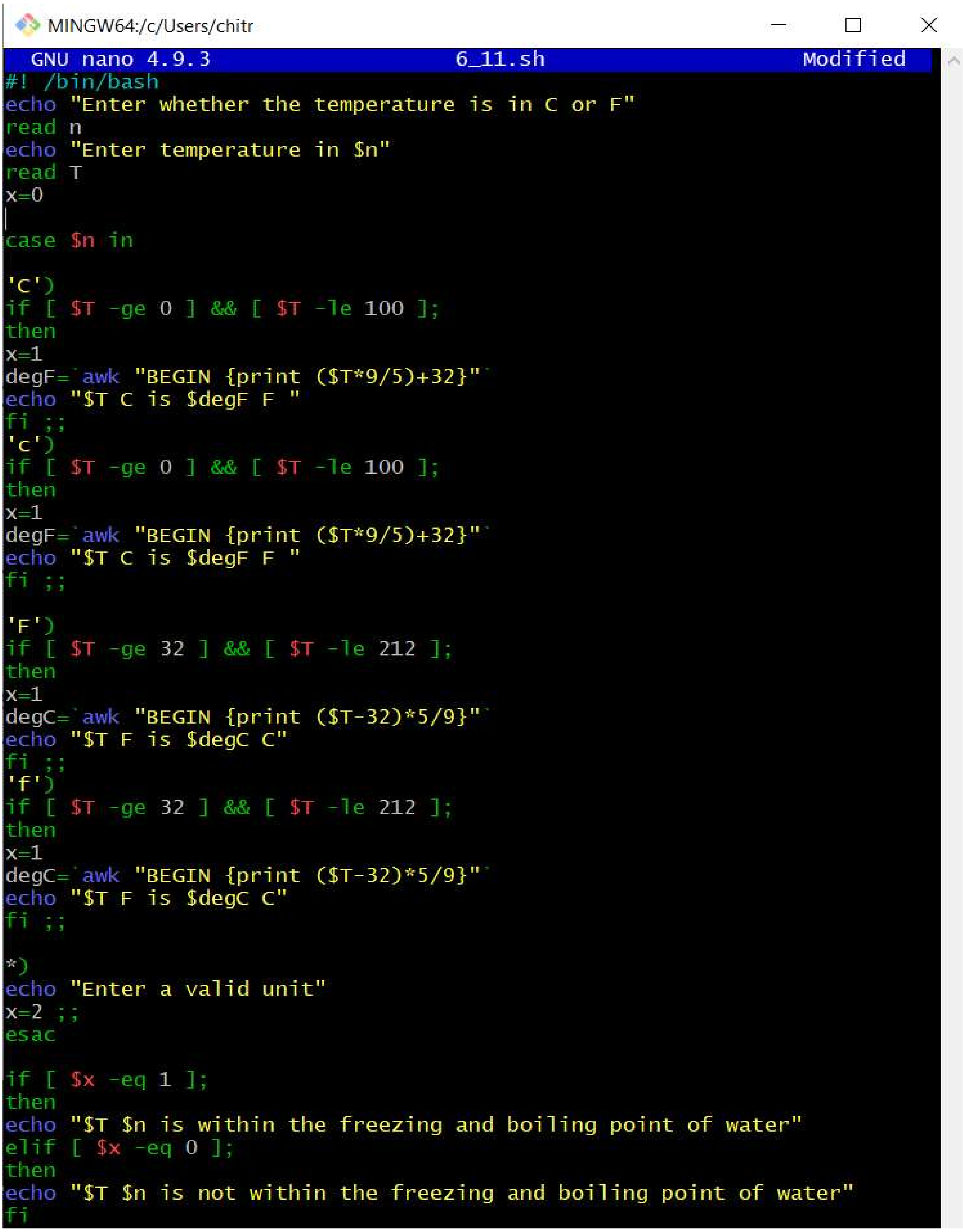


```
192
193
194
195
196
197
196
197
198
199
200
```

Functions

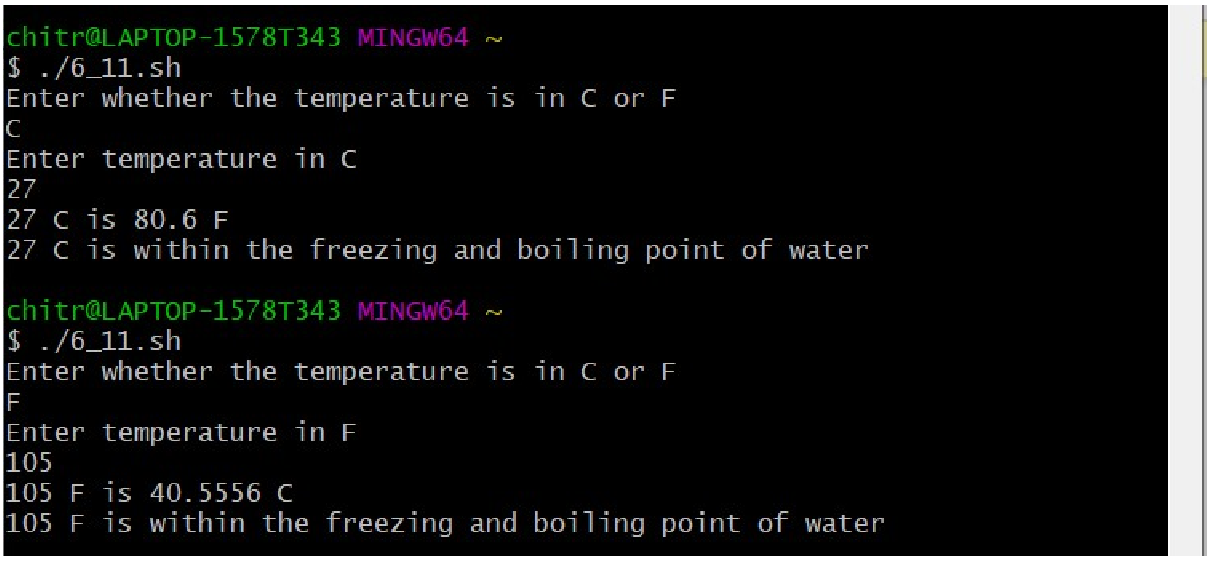
1. Help user find degF or degC based on their Conversion Selection. Use Case Statement and ensure that the inputs are within the Freezing Point (0 °C / 32 °F) and the Boiling Point of Water (100 °C / 212 °F) a. degF = (degC * 9/5) + 32 b. degC = (degF – 32) * 5/9

PROGRAM:



```
  GNU nano 4.9.3                   6_11.sh                        Modified
#! /bin/bash
echo "Enter whether the temperature is in C or F"
read n
echo "Enter temperature in $n"
read T
x=0

case $n in

'C')
if [ $T -ge 0 ] && [ $T -le 100 ];
then
x=1
degF=`awk "BEGIN {print ($T*9/5)+32}"`
echo "$T C is $degF F "
fi ;;
'c')
if [ $T -ge 0 ] && [ $T -le 100 ];
then
x=1
degF=`awk "BEGIN {print ($T*9/5)+32}"`
echo "$T C is $degF F "
fi ;;

'F')
if [ $T -ge 32 ] && [ $T -le 212 ];
then
x=1
degC=`awk "BEGIN {print ($T-32)*5/9}"`
echo "$T F is $degC C"
fi ;;
'f')
if [ $T -ge 32 ] && [ $T -le 212 ];
then
x=1
degC=`awk "BEGIN {print ($T-32)*5/9}"`
echo "$T F is $degC C"
fi ;;

*)
echo "Enter a valid unit"
x=2 ;;
esac

if [ $x -eq 1 ];
then
echo "$T $n is within the freezing and boiling point of water"
elif [ $x -eq 0 ];
then
echo "$T $n is not within the freezing and boiling point of water"
fi
```

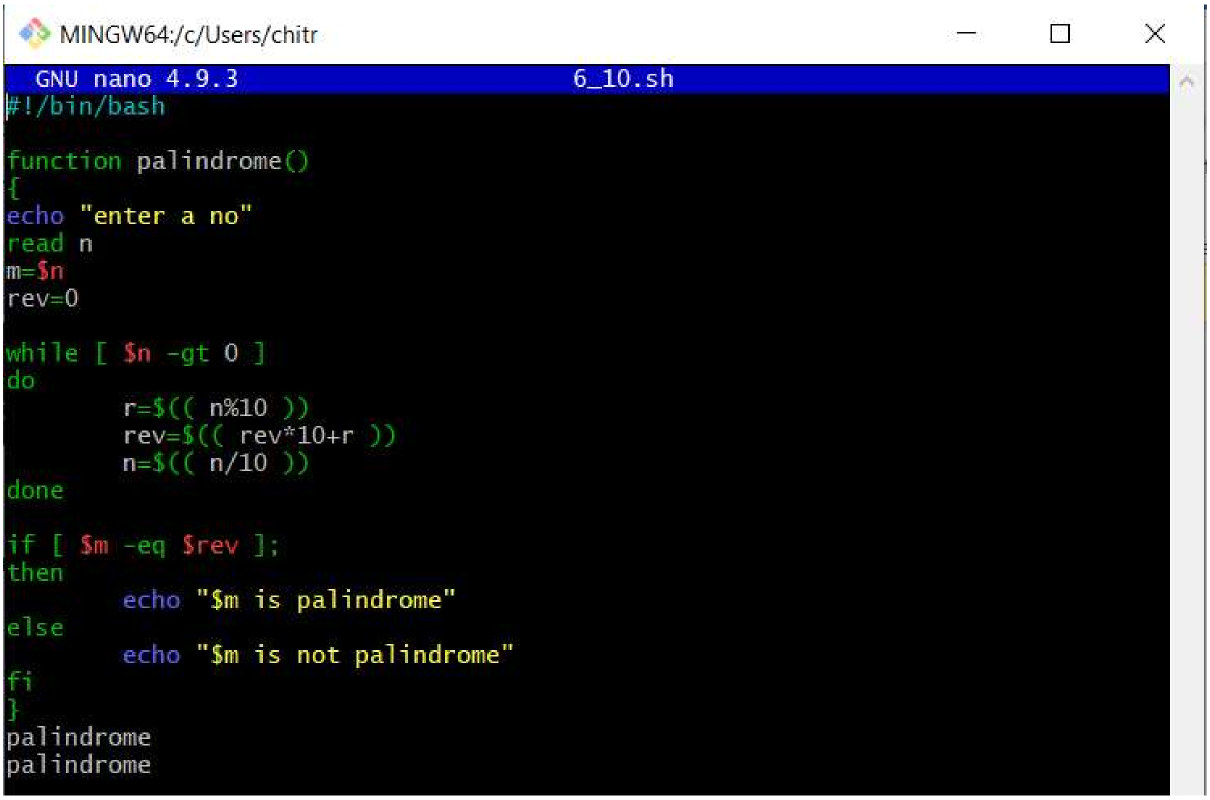
OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_11.sh
Enter whether the temperature is in C or F
C
Enter temperature in C
27
27 C is 80.6 F
27 C is within the freezing and boiling point of water

chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_11.sh
Enter whether the temperature is in C or F
F
Enter temperature in F
105
105 F is 40.5556 C
105 F is within the freezing and boiling point of water
```

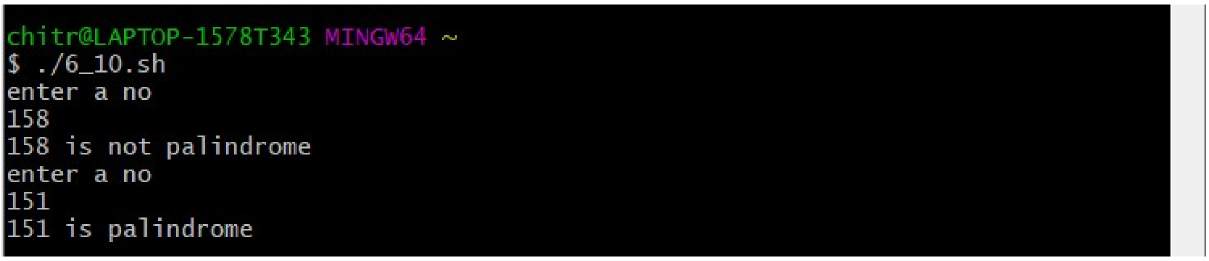
2. Write a function to check if the two numbers are Palindrome

PROGRAM:



```
  GNU nano 4.9.3                    6_10.sh
#!/bin/bash

function palindrome()
{
echo "enter a no"
read n
m=$n
rev=0

while [ $n -gt 0 ]
do
        r=$(( n%10 ))
        rev=$(( rev*10+r ))
        n=$(( n/10 ))
done

if [ $m -eq $rev ];
then
        echo "$m is palindrome"
else
        echo "$m is not palindrome"
fi
}
palindrome
palindrome
```

OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_10.sh
enter a no
158
158 is not palindrome
enter a no
151
151 is palindrome
```

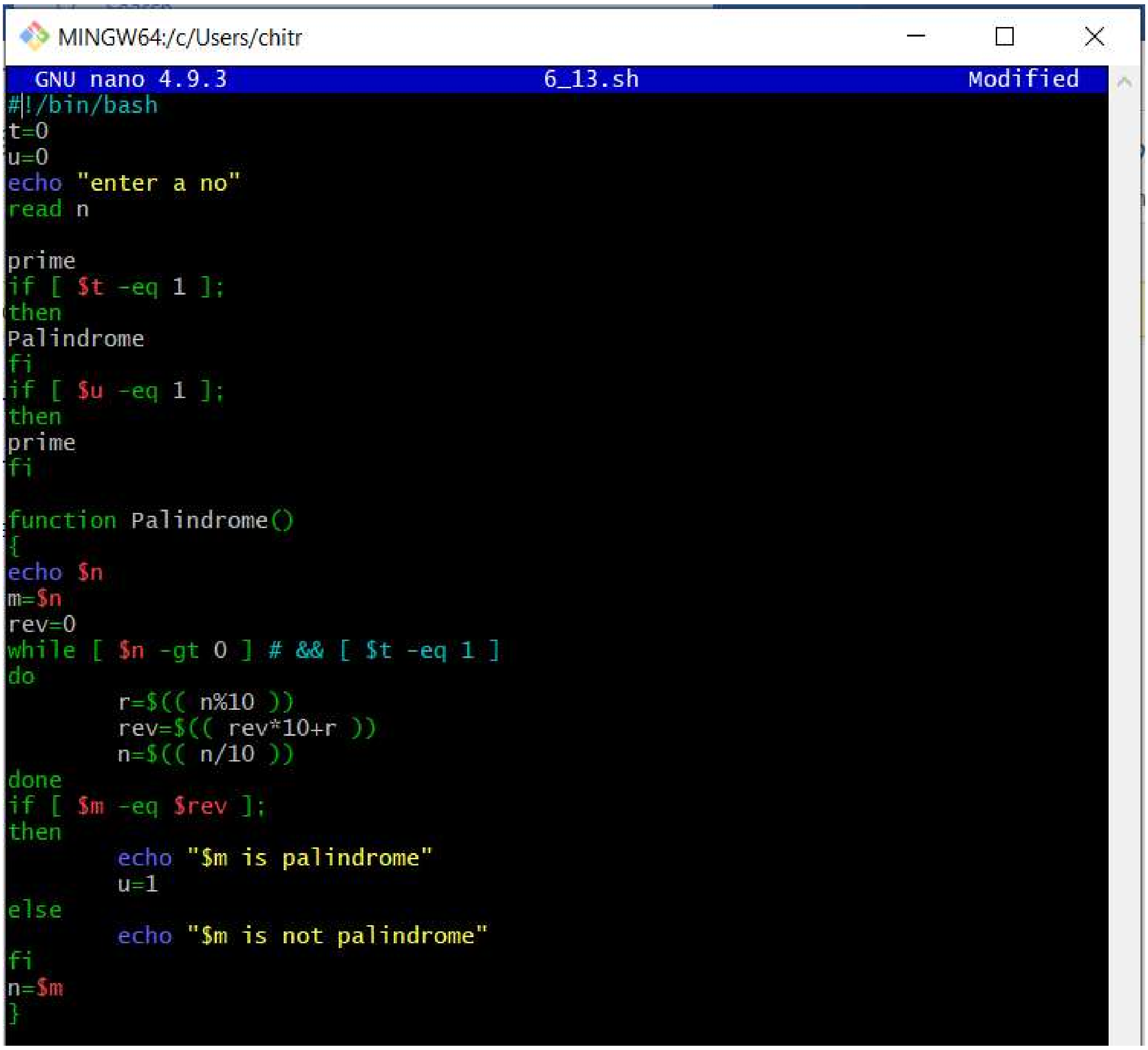
3. Take a number from user and check if the number is a Prime then show that its palindrome is also prime

    a. Write function check if number is Prime

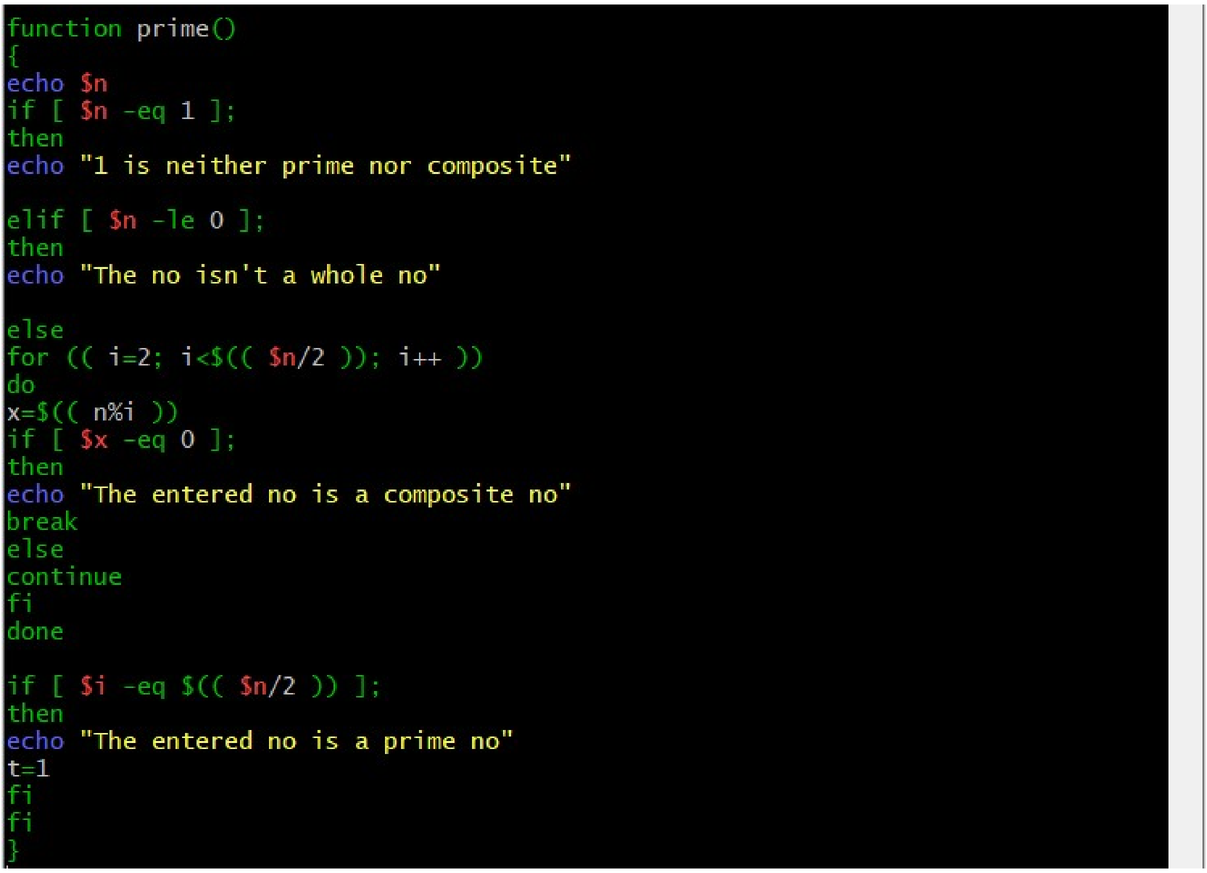
    b. Write function to get the Palindrome.

    c. Check if the Palindrome number is also prime

PROGRAM:

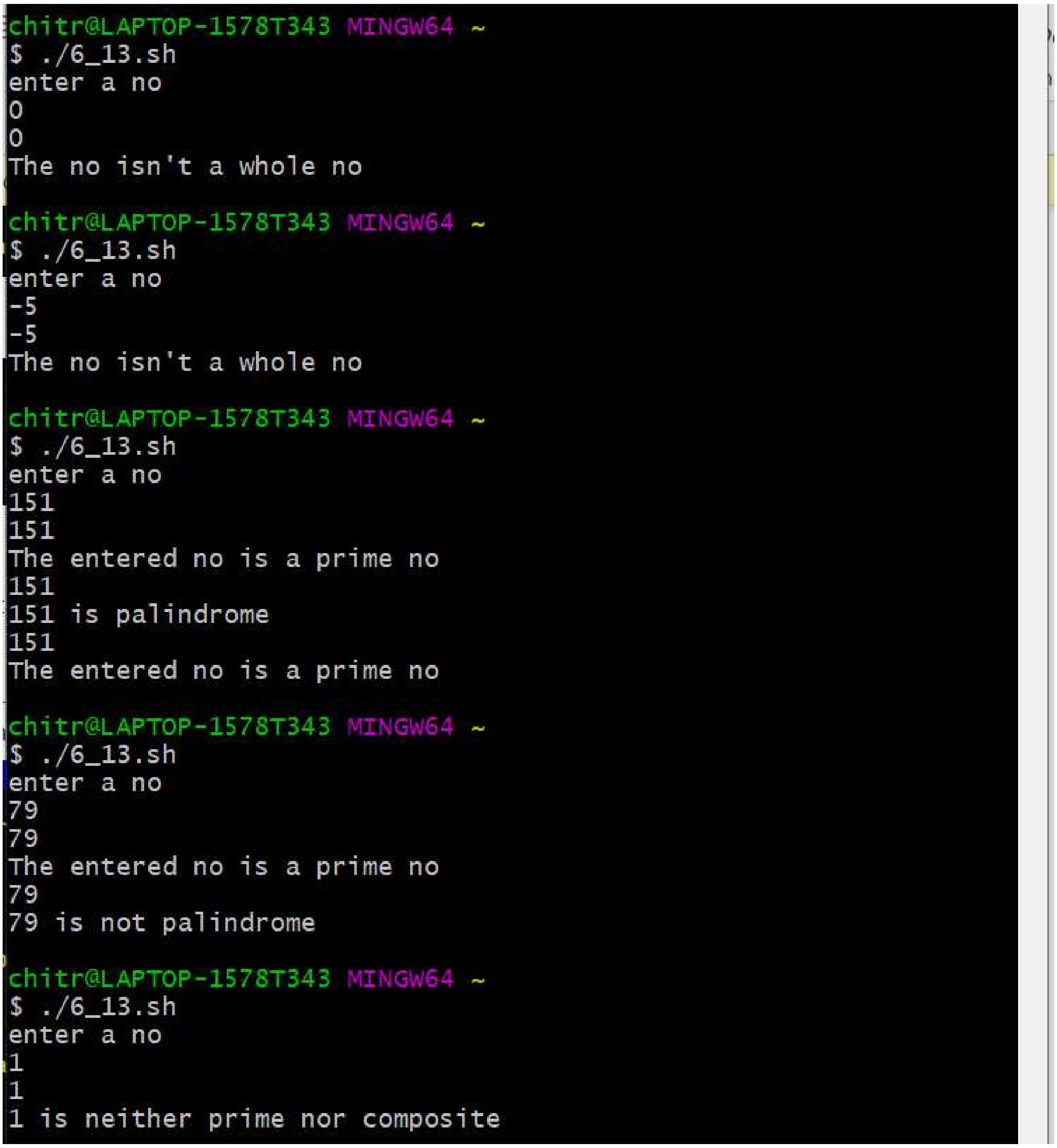


```
  GNU nano 4.9.3                     6_13.sh                        Modified
#!/bin/bash
t=0
u=0
echo "enter a no"
read n

prime
if [ $t -eq 1 ];
then
Palindrome
fi
if [ $u -eq 1 ];
then
prime
fi

function Palindrome()
{
echo $n
m=$n
rev=0
while [ $n -gt 0 ] # && [ $t -eq 1 ]
do
        r=$(( n%10 ))
        rev=$(( rev*10+r ))
        n=$(( n/10 ))
done
if [ $m -eq $rev ];
then
        echo "$m is palindrome"
        u=1
else
        echo "$m is not palindrome"
fi
n=$m
}
```



```
function prime()
{
echo $n
if [ $n -eq 1 ];
then
echo "1 is neither prime nor composite"

elif [ $n -le 0 ];
then
echo "The no isn't a whole no"

else
for (( i=2; i<$(( $n/2 )); i++ ))
do
x=$(( n%i ))
if [ $x -eq 0 ];
then
echo "The entered no is a composite no"
break
else
continue
fi
done

if [ $i -eq $(( $n/2 )) ];
then
echo "The entered no is a prime no"
t=1
fi
fi
}
```

OUTPUT:



```
chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_13.sh
enter a no
0
0
The no isn't a whole no

chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_13.sh
enter a no
-5
-5
The no isn't a whole no

chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_13.sh
enter a no
151
151
The entered no is a prime no
151
151 is palindrome
151
The entered no is a prime no

chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_13.sh
enter a no
79
79
The entered no is a prime no
79
79 is not palindrome

chitr@LAPTOP-1578T343 MINGW64 ~
$ ./6_13.sh
enter a no
1
1
1 is neither prime nor composite
```