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.

Shell code:

##!/bin/bash -x

echo "Enter the value up to you want to print : "

read n

sum=1

for ((i=0;i<n;i++))

do

sum=$((sum \* 2))

echo $sum

done

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./2power.sh

Enter the value up to you want to print :

10

2

4

8

16

32

64

128

256

512

1024

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

Shell code:

##!/bin/bash -x

echo "Enter the number upto you want to print harmonic value : "

read n

sum=0

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

do

div=$(awk 'BEGIN {print '1' / '$i'}' )

sum=$(awk 'BEGIN {print '$sum' + '$div'}')

done

echo "Hormoni value of $n number is $sum"

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./harmonic.sh

Enter the number upto you want to print harmonic value :

10

Hormoni value of 10 number is 2.92897

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

Shell code:

##!/bin/bash -x

echo "Enter the number which you want to check : "

read a

p=0

if (( $a<=1 ))

then

p=1

fi

for ((i=2;i<$a;i++))

do

d=`expr $a % $i`

if(( $d==0 ))

then

p=1

fi

done

if ((p==0))

then

echo "$a is prime number."

else

echo "$a is not prime number"

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./prime.sh

Enter the number which you want to check :

13

13 is prime number.

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./prime.sh

Enter the number which you want to check :

16

16 is not prime number

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

Shell code:

##!/bin/bash -x

echo "Enter the range : "

read b

read a

for((j=$b;j<=$a;j++))

do

p=0

if (( $j<=1 ))

then

p=1

fi

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

do

d=`expr $j % $i`

if(( $d==0 ))

then

p=1

fi

done

if ((p==0))

then

echo "$j"

fi

done

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./primeinrange.sh

Enter the range :

2

20

2

3

5

7

11

13

17

19

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

Shell code:

#!/bin/bash -x

echo "Enter the number factorial : "

read n

fact=1

if [ $n -eq 0 ]

then

fact=1

fi

for ((i=$n;i>0;i--))

do

fact=$(awk 'BEGIN {print '$fact' \* '$i'}')

done

echo "factorial of $n = $fact"

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./factorial.sh

Enter the number factorial :

5

factorial of 5 = 120

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./factorial.sh

Enter the number factorial :

0

factorial of 0 = 1

1. 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.

Shell code:

##!/bin/bash -x

echo " Enter the number to print prime factors : "

read n

echo "Prime factor of $n is : "

for ((i=2;i<=$n;i++))

do

if(($n%i==0))

then

isprime=1

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

do

if(($i%$j==0))

then

isprime=0

break

fi

done

if (($isprime==1))

then

echo $i

fi

fi

done

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./primefactors.sh

Enter the number to print prime factors :

70

Prime factor of 70 is :

2

5

7

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..

Shell code:

##!/bin/bash -x

echo "Enter the value up to you want to print : "

read n

sum=1

for ((i=0;i<n;i++))

do

sum=$((sum \* 2))

echo $sum

if [ $sum -eq 256 ]

then

break

fi

done

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./2powtill256.sh

Enter the value up to you want to print :

15

2

4

8

16

32

64

128

256

1. Find the Magic Number
   1. Ask the user to think of a number n between 1 to 100
   2. Then check with the user if the number is less then n/2 or greater
   3. Repeat till the Magic Number is reached..

Shell code:

##!/bin/bash -x

echo "Think of number n between 1 to 100"

choice=2

start=1

end=100

while [ $choice -ne 1 ]

do

mid=$((($start+$end)/2))

echo "Is your no $mid ? "

echo "1.yes"

echo "2.Greter than this"

echo "3.Less than this"

read choice

if [ $choice -eq 2 ]

then

start=$mid

elif [ $choice -eq 3 ]

then

end=$mid

elif [ $choice -eq 1 ]

then

echo "Magic number is $mid"

else

echo "Invalid number"

fi

done

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./magicnumber.sh

Think of number n between 1 to 100

Is your no 50 ?

1.yes

2.Greter than this

3.Less than this

2

Is your no 75 ?

1.yes

2.Greter than this

3.Less than this

2

Is your no 87 ?

1.yes

2.Greter than this

3.Less than this

3

Is your no 81 ?

1.yes

2.Greter than this

3.Less than this

1

Magic number is 81

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

Shell code:

##!/bin/bash -x

heads=0

tails=0

while (($heads<11 && $tails<11))

do

win=$((RANDOM%2))

if [ $win -eq 0 ]

then

heads=$((heads+1))

else

tails=$((tails+1))

fi

if [ $heads -eq 11 ]

then

echo "Heads Win"

elif [ $tails -eq 11 ]

then

echo "Tails Win"

else

echo "Scorebord"

echo "Heads=$heads Tails=$tails"

fi

done

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./win11.sh

Scorebord

Heads=1 Tails=0

Scorebord

Heads=1 Tails=1

Scorebord

Heads=2 Tails=1

Scorebord

Heads=2 Tails=2

Scorebord

Heads=3 Tails=2

Scorebord

Heads=3 Tails=3

Scorebord

Heads=4 Tails=3

Scorebord

Heads=4 Tails=4

Scorebord

Heads=4 Tails=5

Scorebord

Heads=4 Tails=6

Scorebord

Heads=5 Tails=6

Scorebord

Heads=5 Tails=7

Scorebord

Heads=5 Tails=8

Scorebord

Heads=6 Tails=8

Scorebord

Heads=7 Tails=8

Scorebord

Heads=8 Tails=8

Scorebord

Heads=8 Tails=9

Scorebord

Heads=8 Tails=10

Tails Win

1. 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.

Shell code:

##!/bin/bash -x

start=100

countwin=0

countloss=0

betcount=0

while(($start<200&&$start>0))

do

betcount=$((betcount+1))

win=$((RANDOM%2))

if(($win==0))

then

start=$((start+1))

countwin=$((countwin+1))

else

start=$((start-1))

countloss=$((countloss+1))

fi

if ((start==200))

then

echo "You reaches goal"

elif ((start==0))

then

echo "You broke"

fi

done

echo "Number of won $countwin"

echo "Number of loss $countloss"

echo "Number of bet played $betcount"

per=$(awk 'BEGIN {print '$countwin' / '$betcount' \* '100'}')

lper=$(awk 'BEGIN {print '$countloss' / '$betcount' \* '100'}')

echo "Winning persentage = $per %"

echo "losing persentage = $lper %"

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./gamblergame.sh

You reaches goal

Number of won 3387

Number of loss 3287

Number of bet played 6674

Winning persentage = 50.7492 %

losing persentage = 49.2508 %

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./gamblergame.sh

You broke

Number of won 1531

Number of loss 1631

Number of bet played 3162

Winning persentage = 48.4187 %

losing persentage = 51.5813 %

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 )
   1. degF = (degC \* 9/5) + 32 212
   2. degC = (degF – 32) \* 5/9

Shell code:

#!/bin/bash -x

echo "1.convert Celsius to Fahrenheit"

echo "2.convert Fahrenheit to Celsius"

read a

if((a==1))

then

echo "Enter value in Celsius : "

read c

if((c<=100&&c>=0))

then

t=$(awk 'BEGIN {print '$c' \* '9' / '5'}')

f=$(awk 'BEGIN {print '$t' + '32'}')

echo "$c Celsius = $f Fahrenheit"

else

echo "You should enter value in between 0 to 100"

fi

else

echo "Enter value in Fahrenheit : "

read c

if((c<=212&&c>=32))

then

f=$(awk 'BEGIN {print '$c' - '32'}')

t=$(awk 'BEGIN {print '$f' \* '5' / '9'}')

echo "$c Fahrenheit = $t Celsius"

else

echo "You should enter value in between 32 to 212"

fi

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./FtoC.sh

1.convert Celsius to Fahrenheit

2.convert Fahrenheit to Celsius

1

Enter value in Celsius :

10

10 Celsius = 50 Fahrenheit

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./FtoC.sh

1.convert Celsius to Fahrenheit

2.convert Fahrenheit to Celsius

2

Enter value in Fahrenheit :

36

36 Fahrenheit = 2.22222 Celsius

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

Shell code:

##!/bin/bash -x

echo "Enter number you want to check : "

read n

rev=$n

p=0

while [ $n -ne 0 ]

do

digit=$((n%10))

p=$(((p\*10)+digit))

n=$((n/10))

done

echo "Reverse number is $p"

if [ $rev -eq $p ]

then

echo "Number is palendrome"

else

echo "Number is not palendrome"

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./palendrom.sh

Enter number you want to check :

1221

Reverse number is 1221

Number is palindrome

1. Take a number from user and check if the number is a Prime then show that its palindrome is also prime
   1. Write function check if number is Prime
   2. Write function to get the Palindrome.
   3. Check if the Palindrome number is also prime

Shell code:

##!/bin/bash -x

check\_palinedrome()

{

n=$1

rev=$n

p=0

while [ $n -ne 0 ]

do

digit=$((n%10))

p=$(((p\*10)+digit))

n=$((n/10))

done

echo "Reverse number is $p"

if [ $rev -eq $p ]

then

echo "$1 Number is palendrome"

else

echo "$1 Number is not palendrome"

fi

}

check\_prime()

{

a=$1

p=0

if (( $a<=1 ))

then

p=1

fi

for ((i=2;i<$a;i++))

do

d=`expr $a % $i`

if(( $d==0 ))

then

p=1

fi

done

if ((p==0))

then

echo "$a is prime number."

else

echo "$a is not prime number"

fi

}

echo "Enter the number u want to check : "

read t

check\_prime $t

check\_palinedrome $t

if [ $p -gt 0 ]

then

check\_prime $p

fi

Output :

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./checkprimepalendrom.sh

Enter the number u want to check :

13

13 is prime number.

Reverse number is 31

13 Number is not palendrome

31 is prime number.

suraj@DESKTOP-TFH17A1 MINGW64 ~/codin club/terminal/shfile/day6

$ ./checkprimepalendrom.sh

Enter the number u want to check :

151

151 is prime number.

Reverse number is 151

151 Number is palendrome

151 is prime number.