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

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ cat bash prime.sh

cat: bash: No such file or directory

read -p " enter a prime number " num

for ((i=2;i<=num/2;i++))

do

if [ $(($num%i)) == 0 ]

then

echo " num is not a prime number "

exit

fi

done

echo " num is a prime number "

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x prime.sh

+ read -p ' enter a prime number ' num

enter a prime number 2

+ (( i=2 ))

+ (( i<=num/2 ))

+ echo ' num is a prime number '

num is a prime number

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x prime.sh

+ read -p ' enter a prime number ' num

enter a prime number 3

+ (( i=2 ))

+ (( i<=num/2 ))

+ echo ' num is a prime number '

num is a prime number

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x prime.sh

+ read -p ' enter a prime number ' num

enter a prime number 4

+ (( i=2 ))

+ (( i<=num/2 ))

+ '[' 0 == 0 ']'

+ echo ' num is not a prime number '

num is not a prime number

+ exit

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x prime.sh

+ read -p ' enter a prime number ' num

enter a prime number 5

+ (( i=2 ))

+ (( i<=num/2 ))

+ '[' 1 == 0 ']'

+ (( i++ ))

+ (( i<=num/2 ))

+ echo ' num is a prime number '

num is a prime number

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x prime.sh

+ read -p ' enter a prime number ' num

enter a prime number 20

+ (( i=2 ))

+ (( i<=num/2 ))

+ '[' 0 == 0 ']'

+ echo ' num is not a prime number '

num is not a prime number

+ exit

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash prime.sh

enter a prime number 20

num is not a prime number

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash prime.sh

enter a prime number 30

num is not a prime number

2)WHILE LOOP

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ cat bash headtail.sh

cat: bash: No such file or directory

i=1

while [ $i -lt 12 ]

do

echo $i " head"

echo $i "tail"

((i++))

done

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x headtail.sh

+ i=1

+ '[' 1 -lt 12 ']'

+ echo 1 ' head'

1 head

+ echo 1 tail

1 tail

+ (( i++ ))

+ '[' 2 -lt 12 ']'

+ echo 2 ' head'

2 head

+ echo 2 tail

2 tail

+ (( i++ ))

+ '[' 3 -lt 12 ']'

+ echo 3 ' head'

3 head

+ echo 3 tail

3 tail

+ (( i++ ))

+ '[' 4 -lt 12 ']'

+ echo 4 ' head'

4 head

+ echo 4 tail

4 tail

+ (( i++ ))

+ '[' 5 -lt 12 ']'

+ echo 5 ' head'

5 head

+ echo 5 tail

5 tail

+ (( i++ ))

+ '[' 6 -lt 12 ']'

+ echo 6 ' head'

6 head

+ echo 6 tail

6 tail

+ (( i++ ))

+ '[' 7 -lt 12 ']'

+ echo 7 ' head'

7 head

+ echo 7 tail

7 tail

+ (( i++ ))

+ '[' 8 -lt 12 ']'

+ echo 8 ' head'

8 head

+ echo 8 tail

8 tail

+ (( i++ ))

+ '[' 9 -lt 12 ']'

+ echo 9 ' head'

9 head

+ echo 9 tail

9 tail

+ (( i++ ))

+ '[' 10 -lt 12 ']'

+ echo 10 ' head'

10 head

+ echo 10 tail

10 tail

+ (( i++ ))

+ '[' 11 -lt 12 ']'

+ echo 11 ' head'

11 head

+ echo 11 tail

11 tail

+ (( i++ ))

+ '[' 12 -lt 12 ']'

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash headtail.sh

1 head

1 tail

2 head

2 tail

3 head

3 tail

4 head

4 tail

5 head

5 tail

6 head

6 tail

7 head

7 tail

8 head

8 tail

9 head

9 tail

10 head

10 tail

11 head

11 tail

Check whether the number is palindrome or not using function

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ cat bash palindrome.sh

cat: bash: No such file or directory

echo "Enter the number"

read n

function pal

{

number=$n

reverse=0

while [ $n -gt 0 ]

do

a=`expr $n % 10 `

n=`expr $n / 10 `

reverse=`expr $reverse \\* 10 + $a`

done

echo $reverse

if [ $number -eq $reverse ]

then

echo "Number is palindrome"

else

echo "Number is not palindrome"

fi

}

r=`pal $n`

echo "$r"

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x palindrome.sh

+ echo 'Enter the number'

Enter the number

+ read n

1221

++ pal 1221

++ number=1221

++ reverse=0

++ '[' 1221 -gt 0 ']'

+++ expr 1221 % 10

++ a=1

+++ expr 1221 / 10

++ n=122

+++ expr 0 '\*' 10 + 1

++ reverse=1

++ '[' 122 -gt 0 ']'

+++ expr 122 % 10

++ a=2

+++ expr 122 / 10

++ n=12

+++ expr 1 '\*' 10 + 2

++ reverse=12

++ '[' 12 -gt 0 ']'

+++ expr 12 % 10

++ a=2

+++ expr 12 / 10

++ n=1

+++ expr 12 '\*' 10 + 2

++ reverse=122

++ '[' 1 -gt 0 ']'

+++ expr 1 % 10

++ a=1

+++ expr 1 / 10

++ n=0

+++ expr 122 '\*' 10 + 1

++ reverse=1221

++ '[' 0 -gt 0 ']'

++ echo 1221

++ '[' 1221 -eq 1221 ']'

++ echo 'Number is palindrome'

+ r='1221

Number is palindrome'

+ echo '1221

Number is palindrome'

1221

Number is palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash palindrome.sh

Enter the number

2552

2552

Number is palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash palindrome.sh

Enter the number

32123

32123

Number is palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash palindrome.sh

Enter the number

123456

654321

Number is not palindrome

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

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ cat bash factor.sh

cat: bash: No such file or directory

read -p " enter a number " x

fact=1

for((i=2;i<=x;i++))

{

fact=$((fact \* i))

}

echo $fact

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x factor.sh

+ read -p ' enter a number ' x

enter a number 3

+ fact=1

+ (( i=2 ))

+ (( i<=x ))

+ fact=2

+ (( i++ ))

+ (( i<=x ))

+ fact=6

+ (( i++ ))

+ (( i<=x ))

+ echo 6

6

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x factor.sh

+ read -p ' enter a number ' x

enter a number 5

+ fact=1

+ (( i=2 ))

+ (( i<=x ))

+ fact=2

+ (( i++ ))

+ (( i<=x ))

+ fact=6

+ (( i++ ))

+ (( i<=x ))

+ fact=24

+ (( i++ ))

+ (( i<=x ))

+ fact=120

+ (( i++ ))

+ (( i<=x ))

+ echo 120

120

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash factor.sh

enter a number 6

720

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x factor.sh

+ read -p ' enter a number ' x

enter a number 7

+ fact=1

+ (( i=2 ))

+ (( i<=x ))

+ fact=2

+ (( i++ ))

+ (( i<=x ))

+ fact=6

+ (( i++ ))

+ (( i<=x ))

+ fact=24

+ (( i++ ))

+ (( i<=x ))

+ fact=120

+ (( i++ ))

+ (( i<=x ))

+ fact=720

+ (( i++ ))

+ (( i<=x ))

+ fact=5040

+ (( i++ ))

+ (( i<=x ))

+ echo 5040

5040

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x factor.sh

+ read -p ' enter a number ' x

enter a number 6

+ fact=1

+ (( i=2 ))

+ (( i<=x ))

+ fact=2

+ (( i++ ))

+ (( i<=x ))

+ fact=6

+ (( i++ ))

+ (( i<=x ))

+ fact=24

+ (( i++ ))

+ (( i<=x ))

+ fact=120

+ (( i++ ))

+ (( i<=x ))

+ fact=720

+ (( i++ ))

+ (( i<=x ))

+ echo 720

720

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents/bridgelab

$ bash -x factor.sh

+ read -p ' enter a number ' x

enter a number 10

+ fact=1

+ (( i=2 ))

+ (( i<=x ))

+ fact=2

+ (( i++ ))

+ (( i<=x ))

+ fact=6

+ (( i++ ))

+ (( i<=x ))

+ fact=24

+ (( i++ ))

+ (( i<=x ))

+ fact=120

+ (( i++ ))

+ (( i<=x ))

+ fact=720

+ (( i++ ))

+ (( i<=x ))

+ fact=5040

+ (( i++ ))

+ (( i<=x ))

+ fact=40320

+ (( i++ ))

+ (( i<=x ))

+ fact=362880

+ (( i++ ))

+ (( i<=x ))

+ fact=3628800

+ (( i++ ))

+ (( i<=x ))

+ echo 3628800

3628800

5)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

function highestPowerof2$n()

{

$res = 0;

for ($i = $n; $i >= 1; $i--)

{

// If i is a power of 2

if ((($i & ($i - 1)) == 0)

{

$res = $i;

break;

}

}

return $res;

}

// Driver code

$n = 10;

echo highestPowerof2($n);

OUTPUT;-

Enter the number

4=2^2

6)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

a}

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ cat bash pp.sh

cat: bash: No such file or directory

echo "enter number"

read num

function prime

{

for((i=2; i<=num/2; i++))

do

if [ $((num%i)) -eq 0 ]

then

echo " $num is not a prime number. "

exit

fi

done

echo " $num is a prime number. "

}

r=`prime $number`

echo "$r""r"

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash pp.sh

enter number

5

5 is a prime number.

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash pp.sh

enter number

6

6 is not a prime number.

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash pp.sh

enter number

7

7 is a prime number.

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash pp.sh

enter number

8

8 is not a prime number.

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash pp.sh

enter number

1

1 is a prime number.

B)

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ nano noop.sh

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ nano noop.sh

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash noop.sh

Enter the number

123

321

Number is not palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash noop.sh

Enter the number

5264562

2654625

Number is not palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash noop.sh

Enter the number

123321

123321

Number is palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash noop.sh

Enter the number

963369

963369

Number is palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash -x noop.sh

+ echo 'Enter the number'

Enter the number

+ read n

456654

++ pal 456654

++ number=456654

++ reverse=0

++ '[' 456654 -gt 0 ']'

+++ expr 456654 % 10

++ a=4

+++ expr 456654 / 10

++ n=45665

+++ expr 0 '\*' 10 + 4

++ reverse=4

++ '[' 45665 -gt 0 ']'

+++ expr 45665 % 10

++ a=5

+++ expr 45665 / 10

++ n=4566

+++ expr 4 '\*' 10 + 5

++ reverse=45

++ '[' 4566 -gt 0 ']'

+++ expr 4566 % 10

++ a=6

+++ expr 4566 / 10

++ n=456

+++ expr 45 '\*' 10 + 6

++ reverse=456

++ '[' 456 -gt 0 ']'

+++ expr 456 % 10

++ a=6

+++ expr 456 / 10

++ n=45

+++ expr 456 '\*' 10 + 6

++ reverse=4566

++ '[' 45 -gt 0 ']'

+++ expr 45 % 10

++ a=5

+++ expr 45 / 10

++ n=4

+++ expr 4566 '\*' 10 + 5

++ reverse=45665

++ '[' 4 -gt 0 ']'

+++ expr 4 % 10

++ a=4

+++ expr 4 / 10

++ n=0

+++ expr 45665 '\*' 10 + 4

++ reverse=456654

++ '[' 0 -gt 0 ']'

++ echo 456654

++ '[' 456654 -eq 456654 ']'

++ echo 'Number is palindrome'

+ r='456654

Number is palindrome'

+ echo '456654

Number is palindrome'

456654

Number is palindrome

C)

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ nano c.sh

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash c.sh

Enter the number

123

321

Number is not palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash c.sh

Enter the number

123321

123321

Number is palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ nano c.sh

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash c.sh

Enter the number

213

312

Number is not palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash c.sh

Enter the number

523325

523325

Number is palindrome

Vanishree@LAPTOP-9VLT2MT4 MINGW64 ~/Documents

$ bash -x c.sh

+ echo 'Enter the number'

Enter the number

+ read n

1236321

++ pal 1236321

++ number=1236321

++ reverse=0

++ '[' 1236321 -gt 0 ']'

+++ expr 1236321 % 10

++ a=1

+++ expr 1236321 / 10

++ n=123632

+++ expr 0 '\*' 10 + 1

++ reverse=1

++ '[' 123632 -gt 0 ']'

+++ expr 123632 % 10

++ a=2

+++ expr 123632 / 10

++ n=12363

+++ expr 1 '\*' 10 + 2

++ reverse=12

++ '[' 12363 -gt 0 ']'

+++ expr 12363 % 10

++ a=3

+++ expr 12363 / 10

++ n=1236

+++ expr 12 '\*' 10 + 3

++ reverse=123

++ '[' 1236 -gt 0 ']'

+++ expr 1236 % 10

++ a=6

+++ expr 1236 / 10

++ n=123

+++ expr 123 '\*' 10 + 6

++ reverse=1236

++ '[' 123 -gt 0 ']'

+++ expr 123 % 10

++ a=3

+++ expr 123 / 10

++ n=12

+++ expr 1236 '\*' 10 + 3

++ reverse=12363

++ '[' 12 -gt 0 ']'

+++ expr 12 % 10

++ a=2

+++ expr 12 / 10

++ n=1

+++ expr 12363 '\*' 10 + 2

++ reverse=123632

++ '[' 1 -gt 0 ']'

+++ expr 1 % 10

++ a=1

+++ expr 1 / 10

++ n=0

+++ expr 123632 '\*' 10 + 1

++ reverse=1236321

++ '[' 0 -gt 0 ']'

++ echo 1236321

++ '[' 1236321 -eq 1236321 ']'

++ echo 'Number is palindrome'

+ r='1236321

Number is palindrome'

+ echo '1236321

Number is palindrome'

1236321

Number is palindrome

(6) 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/n function nthHarmonic($N)

function nthHarmonic($N)

{

for ($i = 2; $i=>$N; $i++)

{ $harmonic += (float)1 / $i;

}

return $harmonic;

}

$N = 8; echo nthHarmonic($N)