Repetition Practice Problems With 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.

$ cat table.sh

#!/bin/bash

read -p "Enter a number:" n

for (( counter=1;counter<=$((2\*\*$n));counter++ ))

do

echo $counter

done

$ ./table.sh

Enter a number:3

1

2

3

4

5

6

7

8

$ ./table.sh

+ read -p 'Enter a number:' n

Enter a number:3

+ (( counter=1 ))

+ (( counter<=8 ))

+ echo 1

1

+ (( counter++ ))

+ (( counter<=8 ))

+ echo 2

2

+ (( counter++ ))

+ (( counter<=8 ))

+ echo 3

3

+ (( counter++ ))

+ (( counter<=8 ))

+ echo 4

4

+ (( counter++ ))

+ (( counter<=8 ))

+ echo 5

5

+ (( counter++ ))

+ (( counter<=8 ))

+ echo 6

6

+ (( counter++ ))

+ (( counter<=8 ))

+ echo 7

7

+ (( counter++ ))

+ (( counter<=8 ))

+ echo 8

8

+ (( counter++ ))

+ (( counter<=8 ))

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

$ ./harmonic.sh

+ read -p 'Enter the number: ' num

Enter the number: 4

+ res=0.0

+ i=1

+ '[' 1 -le 4 ']'

+ echo 1/ 1 =

1/ 1 =

++ echo 'scale=2; 0.0 + 1 / 1'

++ bc

./harmonic.sh: line 11: bc: command not found

+ res=

+ (( i++ ))

+ echo 'the sum is : '

the sum is :

+ '[' 2 -le 4 ']'

+ echo 1/ 2 =

1/ 2 =

++ echo 'scale=2; + 1 / 2'

++ bc

./harmonic.sh: line 11: bc: command not found

+ res=

+ (( i++ ))

+ echo 'the sum is : '

the sum is :

+ '[' 3 -le 4 ']'

+ echo 1/ 3 =

1/ 3 =

++ echo 'scale=2; + 1 / 3'

++ bc

./harmonic.sh: line 11: bc: command not found

+ res=

+ (( i++ ))

+ echo 'the sum is : '

the sum is :

+ '[' 4 -le 4 ']'

+ echo 1/ 4 =

1/ 4 =

++ echo 'scale=2; + 1 / 4'

++ bc

./harmonic.sh: line 11: bc: command not found

+ res=

+ (( i++ ))

+ echo 'the sum is : '

the sum is :

+ '[' 5 -le 4 ']'

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

$ cat primeno.sh

#!/bin/bash

echo -n "Enter the number:"no

read no

i=2

for (( i=2;i<=no/2;i++ ))

do

if [ $((no%i)) -eq 0 ]

then

echo "$no is not prime number"

exit

fi

done

echo "$num is prime number"

$ ./primeno.sh

Enter the number:no13

is prime number

$ ./primeno.sh

Enter the number:no5

is prime number

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

$ cat primenorangef.sh

#!/bin/bash

echo 'Enter no'

read x

n=2

while [ $n -le $x ]

do

i=2

count=1

while [ $i -lt $n ]

do

if [ `expr $n % $i` -eq 0 ]

then

count=0

break

fi

i=`expr $i + 1`

done

if [ $count -eq 1 ]

then

echo "$n is Prime"

fi

n=`expr $n + 1`

done

$ ./primenorangef.sh

+ echo 'Enter no'

Enter no

+ read x

2

+ n=2

+ '[' 2 -le 2 ']'

+ i=2

+ count=1

+ '[' 2 -lt 2 ']'

+ '[' 1 -eq 1 ']'

+ echo '2 is Prime'

2 is Prime

++ expr 2 + 1

+ n=3

+ '[' 3 -le 2 ']'

$ ./primenorangef.sh

Enter no

10

2 is Prime

3 is Prime

5 is Prime

7 is Prime

5 write a program that computes a factorial of a number takes an input.

$ cat factor.sh

#!/bin/bash

echo "Enter a number"

read num

fact=1

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

do

fact=$(( fact \* i ))

echo $fact

done

$ ./factor.sh

Enter a number

5

1

2

6

24

120

6 write a program to compute factorial number N using prime factorization method

$ cat primefactor.sh

#!/bin/bash -x

# find prime factors using while and for loop

read -p "enter the input :" num

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

while [ $(($num % i)) -eq 0 ];

do

echo $i

num=$(( $num / i ))

done

done

echo " "

$ ./primefactor.sh

+ read -p 'enter the input :' num

enter the input :5

+ (( i=2 ))

+ (( i<=num ))

+ '[' 1 -eq 0 ']'

+ (( i++ ))

+ (( i<=num ))

+ '[' 2 -eq 0 ']'

+ (( i++ ))

+ (( i<=num ))

+ '[' 1 -eq 0 ']'

+ (( i++ ))

+ (( i<=num ))

+ '[' 0 -eq 0 ']'

+ echo 5

5

+ num=1

+ '[' 1 -eq 0 ']'

+ (( i++ ))

+ (( i<=num ))

Rapitation practice problems with while loop

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

$ cat primenorange\_while.sh

#!/bin/bash -x

read -p "Enter the nth number:" number

counter=0

result=0

while (( $counter<=$number && $result<256 ))

do

result=$((2\*\*counter))

echo $result

counter=$((counter+1))

done

$ ./primenorange.sh

+ read -p 'Enter the nth number:' number

Enter the nth number:4

+ counter=0

+ result=0

+ (( 0<=4 && 0<256 ))

+ result=1

+ echo 1

1

+ counter=1

+ (( 1<=4 && 1<256 ))

+ result=2

+ echo 2

2

+ counter=2

+ (( 2<=4 && 2<256 ))

+ result=4

+ echo 4

4

+ counter=3

+ (( 3<=4 && 4<256 ))

+ result=8

+ echo 8

8

+ counter=4

+ (( 4<=4 && 8<256 ))

+ result=16

+ echo 16

16

+ counter=5

+ (( 5<=4 && 16<256 ))

3 extend the flip coin problem till either heads or tails wins 11 times

$ cat headtail\_while.sh

#!/bin/bash -x

flips=0

heads=0

tails=0

while [ $flips -lt 11 ]

do

Result=$((RANDOM%2))

if [ ${Result} -eq 0 ]

then

((heads++))

elif [ ${Result} -eq 1 ]

then

((tails++))

fi

((flips++))

done

echo "You got 11 head and 11 tails"

$ ./headtail\_while.sh

+ flips=0

+ heads=0

+ tails=0

+ '[' 0 -lt 11 ']'

+ Result=0

+ '[' 0 -eq 0 ']'

+ (( heads++ ))

+ (( flips++ ))

+ '[' 1 -lt 11 ']'

+ Result=0

+ '[' 0 -eq 0 ']'

+ (( heads++ ))

+ (( flips++ ))

+ '[' 2 -lt 11 ']'

+ Result=1

+ '[' 1 -eq 0 ']'

+ '[' 1 -eq 1 ']'

+ (( tails++ ))

+ (( flips++ ))

+ '[' 3 -lt 11 ']'

+ Result=0

+ '[' 0 -eq 0 ']'

+ (( heads++ ))

+ (( flips++ ))

+ '[' 4 -lt 11 ']'

+ Result=0

+ '[' 0 -eq 0 ']'

+ (( heads++ ))

+ (( flips++ ))

+ '[' 5 -lt 11 ']'

+ Result=1

+ '[' 1 -eq 0 ']'

+ '[' 1 -eq 1 ']'

+ (( tails++ ))

+ (( flips++ ))

+ '[' 6 -lt 11 ']'

+ Result=1

+ '[' 1 -eq 0 ']'

+ '[' 1 -eq 1 ']'

+ (( tails++ ))

+ (( flips++ ))

+ '[' 7 -lt 11 ']'

+ Result=1

+ '[' 1 -eq 0 ']'

+ '[' 1 -eq 1 ']'

+ (( tails++ ))

+ (( flips++ ))

+ '[' 8 -lt 11 ']'

+ Result=0

+ '[' 0 -eq 0 ']'

+ (( heads++ ))

+ (( flips++ ))

+ '[' 9 -lt 11 ']'

+ Result=1

+ '[' 1 -eq 0 ']'

+ '[' 1 -eq 1 ']'

+ (( tails++ ))

+ (( flips++ ))

+ '[' 10 -lt 11 ']'

+ Result=0

+ '[' 0 -eq 0 ']'

+ (( heads++ ))

+ (( flips++ ))

+ '[' 11 -lt 11 ']'

+ echo 'You got 11 head and 11 tails'

You got 11 head and 11 tails

4 write a program where a gambler starts with RS100 and places RS1 bet unit 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

$ cat gamble.sh

#! /bin/bash -x

gambler\_money=100

is\_bet=1

bet\_times=0

bet\_wons=0

bet\_lose=0

while [ $gambler\_money -gt 0 ] && [ $gambler\_money -lt 200 ] && [ $is\_bet -eq 1 ]

do

flip\_coin=$((RANDOM%2))

if [ $flip\_coin -eq 0 ]

then

bet\_wons=$((bet\_wons+1))

else

gambler\_money=$((gambler\_money-1))

bet\_lose=$((bet\_lose+1))

fi

bet\_times=$((bet\_times+1))

done

echo "Money:$gambler\_money , BetTimes:$bet\_times, Won:$bet\_wons,Lose:$bet\_lose"

$ ./gamble.sh

+ gambler\_money=100

+ is\_bet=1

+ bet\_times=0

+ bet\_wons=0

+ bet\_lose=0

+ '[' 100 -gt 0 ']'

+ '[' 100 -lt 200 ']'

+ '[' 1 -eq 1 ']'

+ flip\_coin=0

+ '[' 0 -eq 0 ']'

+ bet\_wons=1

+ bet\_times=1

+ '[' 100 -gt 0 ']'

+ '[' 100 -lt 200 ']'

+ '[' 1 -eq 1 ']'

+ flip\_coin=0

+ '[' 0 -eq 0 ']'

+ bet\_wons=2

+ bet\_times=2

+ '[' 100 -gt 0 ']'

+ '[' 100 -lt 200 ']'

+ '[' 1 -eq 1 ']'

+ flip\_coin=0

+ '[' 0 -eq 0 ']'

+ bet\_wons=3

+ bet\_times=3

+ '[' 100 -gt 0 ']'

+ '[' 100 -lt 200 ']'

+ '[' 1 -eq 1 ']'

+ flip\_coin=0

+ '[' 0 -eq 0 ']'

+ bet\_wons=4

+ bet\_times=4

Upto

+ '[' 1 -gt 0 ']'

+ '[' 1 -lt 200 ']'

+ '[' 1 -eq 1 ']'

+ flip\_coin=1

+ '[' 1 -eq 0 ']'

+ gambler\_money=0

+ bet\_lose=100

+ bet\_times=228

+ '[' 0 -gt 0 ']'

+ echo 'Money:0 , BetTimes:228, Won:128,Lose:100'

Money:0 , BetTimes:228, Won:128,Lose:100

Function practice problems

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)

$ cat tempconv.sh

#!/bin/bash

#temperature conversion using function and switch cases

read -p "Enter the tenperature" temp

function temp\_con\_cel()

{

degC=$(( ($temp-32)\*5/9 ))

echo "The temperature in" $degC "degrees"

}

function temp\_con\_fara()

{

degF=$(( ($temp\*9/5)+32 ))

echo "The temperature in " $degF "degrees"

}

echo "F.Fahrenheit"

echo "C.Celsius"

read ch

case $ch in

F) temp\_con\_fara;;

C) temp\_con\_cel;;

esac

$ ./tempconv.sh

+ read -p 'Enter the tenperature' temp

Enter the tenperature28

+ echo F.Fahrenheit

F.Fahrenheit

+ echo C.Celsius

C.Celsius

+ read ch

F

+ case $ch in

+ temp\_con\_fara

+ degF=82

+ echo 'The temperature in ' 82 degrees

The temperature in 82 degrees

$ ./tempconv.sh

+ read -p 'Enter the tenperature' temp

Enter the tenperature34

+ echo F.Fahrenheit

F.Fahrenheit

+ echo C.Celsius

C.Celsius

+ read ch

C

+ case $ch in

+ temp\_con\_cel

+ degC=1

+ echo 'The temperature in' 1 degrees

The temperature in 1 degrees

2 write a program to check if the two numbers are palindromes

$ cat palindromes.sh

#! /bin/bash -x

echo "Enter the number"

read nun

s=0

rev=""

temp=$num

while [ $num -gt 0 ]

do

s=$(( $num %10 ))

num=$(( $num/10 ))

rev=$( echo ${rev}${s} )

done

if [ $temp -eq $rev ]

then

echo "Number is palindrome"

else

echo "number is not palindrome"

fi

$ ./palindromes.sh

+ echo 'Enter the number'

Enter the number

+ read num

234

+ s=0

+ rev=

+ temp=234

+ '[' 234 -gt 0 ']'

+ s=4

+ num=23

++ echo 4

+ rev=4

+ '[' 23 -gt 0 ']'

+ s=3

+ num=2

++ echo 43

+ rev=43

+ '[' 2 -gt 0 ']'

+ s=2

+ num=0

++ echo 432

+ rev=432

+ '[' 0 -gt 0 ']'

+ '[' 234 -eq 432 ']'

+ echo 'number is not palindrome'

number is not palindrome

$ ./palindromes.sh

+ echo 'Enter the number'

Enter the number

+ read num

252

+ s=0

+ rev=

+ temp=252

+ '[' 252 -gt 0 ']'

+ s=2

+ num=25

++ echo 2

+ rev=2

+ '[' 25 -gt 0 ']'

+ s=5

+ num=2

++ echo 25

+ rev=25

+ '[' 2 -gt 0 ']'

+ s=2

+ num=0

++ echo 252

+ rev=252

+ '[' 0 -gt 0 ']'

+ '[' 252 -eq 252 ']'

+ echo 'Number is palindrome'

Number is palindrome

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

$ cat primepalindrome.sh

#!/bin/bash -x

function check\_prime()

{

count=0

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

do

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

then

echo "Number is prime"

break

else

echo "number is not prime"

break

fi

done

}

read -p "Enter the number " num

check\_prime "$num"

function check\_palindrom()

{

rem=0

rev=""

temp=$num

while [ $num -gt 0 ]

do

s=$(( $num % 10 ))

num=$(( $num / 10 ))

rev=$( echo ${rev}${s} )

done

if [ $temp -eq $rev ];

then

echo "Number is palindrome"

else

echo "Number is NOT palindrome"

fi

}

check\_palindrom "$num"

$ ./primepalindrome.sh

+ read -p 'Enter the number ' num

Enter the number 3

+ check\_prime 3

+ count=0

+ (( i=2 ))

+ (( i<=num/2 ))

+ check\_palindrom 3

+ rem=0

+ rev=

+ temp=3

+ '[' 3 -gt 0 ']'

+ s=3

+ num=0

++ echo 3

+ rev=3

+ '[' 0 -gt 0 ']'

+ '[' 3 -eq 3 ']'

+ echo 'Number is palindrome'

Number is palindrome