**Stage\_2\_Assignment\_2**

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.

**Code:**

#!/bin/bash

read -p "Enter the value for n: " n

for (( c=1; c<=$n; c++ ))

do

r=$(( 2 \*\* $c))

echo "2 ^ " $c "= " $r

done

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c1.sh

Enter the value for n: 5

2 ^ 1 = 2

2 ^ 2 = 4

2 ^ 3 = 8

2 ^ 4 = 16

2 ^ 5 = 32

**Code**

#!/bin/bash -x

read -p "Enter the value for n: " n

for (( c=1; c<=$n; c++ ))

do

r=$(( 2 \* $c))

echo "2 x " $c "= " $r

done

**Output**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/for\_loop

$ ./q1.sh

+ read -p 'Enter the value for n: ' n

Enter the value for n: 5

+ (( c=1 ))

+ (( c<=5 ))

+ r=2

+ echo '2 x ' 1 '= ' 2

2 x 1 = 2

+ (( c++ ))

+ (( c<=5 ))

+ r=4

+ echo '2 x ' 2 '= ' 4

2 x 2 = 4

+ (( c++ ))

+ (( c<=5 ))

+ r=6

+ echo '2 x ' 3 '= ' 6

2 x 3 = 6

+ (( c++ ))

+ (( c<=5 ))

+ r=8

+ echo '2 x ' 4 '= ' 8

2 x 4 = 8

+ (( c++ ))

+ (( c<=5 ))

+ r=10

+ echo '2 x ' 5 '= ' 10

2 x 5 = 10

+ (( c++ ))

+ (( c<=5 ))

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



**Code:**

#!/bin/bash -x

read -p "Enter the value for n: " n

sum=0

for (( c=1; c<=$n; c++ ))

do

r=$(awk "BEGIN{print 1 / $c}")

sum=$(awk "BEGIN{print $sum + $r}")

done

echo "The" $n"th harmonic number is: "$sum

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/for\_loop

$ ./q2.sh

+ read -p 'Enter the value for n: ' n

Enter the value for n: 3

+ sum=0

+ (( c=1 ))

+ (( c<=3 ))

++ awk 'BEGIN{print 1 / 1}'

+ r=1

++ awk 'BEGIN{print 0 + 1}'

+ sum=1

+ (( c++ ))

+ (( c<=3 ))

++ awk 'BEGIN{print 1 / 2}'

+ r=0.5

++ awk 'BEGIN{print 1 + 0.5}'

+ sum=1.5

+ (( c++ ))

+ (( c<=3 ))

++ awk 'BEGIN{print 1 / 3}'

+ r=0.333333

++ awk 'BEGIN{print 1.5 + 0.333333}'

+ sum=1.83333

+ (( c++ ))

+ (( c<=3 ))

+ echo The '3th harmonic number is: 1.83333'

The 3th harmonic number is: 1.83333

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

**Code:**

#!/bin/bash

read -p "Enter the value for number: " n

flag=0

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

do

if (( ( $n == 2 ) ))

then

flag=0

else

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

then

flag=1

fi

fi

done

if [ $flag -eq 0 ]

then

echo $n " is prime"

else

echo $n " is not prime"

fi

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c3.sh

Enter the value for number: 3

3 is prime

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c3.sh

Enter the value for number: 4

4 is not prime

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

**Code:**

#!/bin/bash

read -p "Enter the value for range: " n

for(( r=2; r<=n; r++ ))

do

flag=0

for (( d=2; d<$r; d++ ))

do

if (( ( $r == 2 ) ))

then

flag=0

else

if (( ($r % $d == 0 ) ))

then

flag=1

fi

fi

done

if [ $flag -eq 0 ]

then

echo $r " is prime"

fi

done

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c3.sh

Enter the value for range: 10

2 is prime

3 is prime

5 is prime

7 is prime

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

Factorial – 5! = 1 \* 2 \* 3 \* 4 \* 5

**Code:**

#!/bin/bash

read -p "Enter the value for range: " n

f=1

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

do

f=$(( $f \* $i ))

done

echo "Factorial of $n is" $f

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/for\_loop

$ ./q5.sh

Enter the value for range: 4

Factorial of 4 is 24

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/for\_loop

$ ./q5.sh

Enter the value for range: 5

Factorial of 5 is 120

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.

**Code:**

#!/bin/bash -x

read -p "Enter the value for range: " n

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

do

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

then

if (( ( $i % 2 != 0 | $i == 2 ) ))

then

echo $i " is a factor"

fi

fi

done

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/for\_loop

$ ./q6.sh

+ read -p 'Enter the value for range: ' n

Enter the value for range: 10

+ (( i=2 ))

+ (( i<n ))

+ (( ( 10 % 2 == 0 ) ))

+ (( ( 2 % 2 != 0 | 2 == 2 ) ))

+ echo 2 ' is a factor'

2 is a factor

+ (( i++ ))

+ (( i<n ))

+ (( ( 10 % 3 == 0 ) ))

+ (( i++ ))

+ (( i<n ))

+ (( ( 10 % 4 == 0 ) ))

+ (( i++ ))

+ (( i<n ))

+ (( ( 10 % 5 == 0 ) ))

+ (( ( 5 % 2 != 0 | 5 == 2 ) ))

+ echo 5 ' is a factor'

5 is a factor

+ (( i++ ))

+ (( i<n ))

+ (( ( 10 % 6 == 0 ) ))

+ (( i++ ))

+ (( i<n ))

+ (( ( 10 % 7 == 0 ) ))

+ (( i++ ))

+ (( i<n ))

+ (( ( 10 % 8 == 0 ) ))

+ (( i++ ))

+ (( i<n ))

+ (( ( 10 % 9 == 0 ) ))

+ (( i++ ))

+ (( i<n ))

Repetition Practice Problems with 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..

**Code:**

#!/bin/bash

read -p "Enter the value for n: " n

c=1

while (( $c <= $n ))

do

r=$(( 2 \*\* $c))

if [ $r -le 256 ]

then

echo "2 ^ " $c "= " $r

fi

((c++))

done

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/while\_loop

$ ./q1.sh

Enter the value for n: 10

2 ^ 1 = 2

2 ^ 2 = 4

2 ^ 3 = 8

2 ^ 4 = 16

2 ^ 5 = 32

2 ^ 6 = 64

2 ^ 7 = 128

2 ^ 8 = 256

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..

**Code**

#! /bin/bash

flag=0

low=1

high=100

mid=$(( ( $low + $high )/2 ))

i=$low

while (( i<=$high ))

do echo "low = "$low

echo "high = "$high

echo "mid = " $mid

echo "Is the value less than mid Press 0"

echo "Is the value greater than mid Press 1"

read -p "enter your option " ch

if [ $ch -eq 0 ]

then

low=$low

high=$mid

mid=$(( ( $low + $high )/2 ))

elif [ $ch -eq 1 ]

then

#low=$(( $high + 1 ))

low=$(( $mid + 1 ))

mid=$(( ( $low + $high ) / 2 ))

else

echo "I think i found the number its $mid"

flag=1

fi

if [ $flag -eq 1 ]

then

break

fi

((i++))

done

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c5.sh

low = 1

high = 100

mid = 50

Is the value less than mid Press 0

Is the value greater than mid Press 1

enter your option 0

low = 1

high = 50

mid = 25

Is the value less than mid Press 0

Is the value greater than mid Press 1

enter your option 0

low = 1

high = 25

mid = 13

Is the value less than mid Press 0

Is the value greater than mid Press 1

enter your option 0

low = 1

high = 13

mid = 7

Is the value less than mid Press 0

Is the value greater than mid Press 1

enter your option 1

low = 8

high = 13

mid = 10

Is the value less than mid Press 0

Is the value greater than mid Press 1

enter your option 4

I think i found the number its 10

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

**Code:**

#!/bin/bash

flip=$((RANDOM % 2))

f=0

h=0

t=0

while (( $f == 0 ))

do

flip=$((RANDOM % 2))

if [ $flip -eq 1 ]

then

((h++))

if [ $h -eq 11 ]

then

f=1

echo "Heads wins"

fi

else

((t++))

if [ $t -eq 11 ]

then

f=1

echo "Tails wins"

fi

fi

done

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/while\_loop

$ ./q2.sh

Heads wins

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/while\_loop

$ ./q2.sh

Heads wins

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/while\_loop

$ ./q2.sh

Tails wins

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.

**Code:**

stake=100

goal=200

wins=0

loose=0

while (( $stake >= 1 ))

do

bet=$(( RANDOM % 2 ))

if [ $bet -eq 1 ]

then

((wins++))

stake=$(( $stake + 10 ))

else

((loose++))

stake=$(( $stake - 10 ))

fi

if (( $stake < 1 ))

then

echo "i have gone broke with " $loose "losses"

break

fi

if (( ( $stake == $goal ) ))

then

echo "I have reached my goal with " $wins "Wins"

break

fi

done

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c6.sh

I have reached my goal with 22 Wins

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c6.sh

I have reached my goal with 48 Wins

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./c6.sh

i have gone broke with 122 losses

Functions 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 )

a. degF = (degC \* 9/5) + 32

b. degC = (degF – 32) \* 5/9

**code:**

#!/bin/bash

echo "Choose the conversion type"

echo "1. Centigrade to Farenheit"

echo "2. Farenheit to Centigrade"

read -p "Enter your choice: " ch

C\_to\_F()

{

read -p "Enter the value in centigrade: " degC

if (( ( $degC >= 0 & $degC <= 100 ) ))

then

F=$(awk "BEGIN{ print ($degC \* 9 / 5 ) + 32 }")

echo $degC " Centigrade = " $F " Farenheit"

else

echo"Enter proper value"

fi

}

F\_to\_C()

{

read -p "Enter the value in Farenheit: " degF

if (( ( $degF >= 32 & $degF <= 212 ) ))

then

C=$(awk "BEGIN{ print ($degF - 32) \* (5 / 9) }")

echo $degF " Farenheit = " $C " Centigrade"

else

echo"Enter proper value"

fi

}

case "$ch" in

"1") C\_to\_F

;;

"2") F\_to\_C

;;

\*) echo "Enter proper choice"

;;

esac

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q3.sh

Choose the conversion type

1. Centigrade to Farenheit

2. Farenheit to Centigrade

Enter your choice: 10

Enter proper choice

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q3.sh

Choose the conversion type

1. Centigrade to Farenheit

2. Farenheit to Centigrade

Enter your choice: 1

Enter the value in centigrade: 10

10 Centigrade = 50 Farenheit

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q3.sh

Choose the conversion type

1. Centigrade to Farenheit

2. Farenheit to Centigrade

Enter your choice: 2

Enter the value in Farenheit: 50

50 Farenheit = 10 Centigrade

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

**Code:**

#! /bin/bash

palin()

{

if [ $1 -eq $2 ]

then

echo $temp " and " $2 " is palindrome of each other"

else

echo $temp " and " $2 " is not palindrome of each other"

fi

}

read -p "Enter the number1 u want to check for: " n1

read -p "Enter the number2 which is to check is palindrome of number1: " n2

temp=$n1

sum=0

while [ $n1 != 0 ]

do

r=$(($n1 % 10))

sum=$(( ( ($sum \* 10) + $r) ))

n1=$(($n1 / 10))

done

#echo $sum

palin $sum $n2

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q1.sh

Enter the number1 u want to check for: 1234

Enter the number2 which is to check is palindrome of number1: 121

1234 and 121 is not palindrome of each other

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q1.sh

Enter the number1 u want to check for: 1234

Enter the number2 which is to check is palindrome of number1: 4321

1234 and 4321 is palindrome of each other

**Code:**

#! /bin/bash

palin()

{

if [ $1 -eq $2 ]

then

echo $1 "is palindrome"

else

echo $1 "is not palindrome"

fi

}

read -p "Enter the number u want to check for: " n

temp=$n

sum=0

while [ $n != 0 ]

do

r=$(($n % 10))

sum=$(( ( ($sum \* 10) + $r) ))

n=$(($n / 10))

done

#echo $sum

palin $temp $sum

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q1.sh

Enter the number u want to check for: 121

121 is palindrome

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q1.sh

Enter the number u want to check for: 123

123 is not 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

**Code:**

#! /bin/bash

flag=0

flag2=0

palin()

{

temp=$1

sum=0

while [ $temp != 0 ]

do

r=$(($temp % 10))

sum=$(( ( ($sum \* 10) + $r) ))

temp=$(($temp / 10))

done

checkPrime $sum

}

checkPrime()

{ temp2=$1

if (( ( $1 <= 2 & $1 != 0 ) ))

then

echo $1 "is not prime"

else

for (( d=2; d<$1; d++ ))

do

if (( ($1 % $d == 0 ) ))

then

flag=1

flag2=1

fi

done

fi

}

read -p "Enter the number u want to check the prime and its palindrome prime property for: " n

checkPrime $n

if [ $flag -eq 0 ]

then

echo $n " is prime"

palin $n

if [ $flag2 -eq 0 ]

then

echo "its palindrome is also prime"

else

echo "its palindrome is not prime"

fi

else

echo $n " is not prime"

fi

**Output:**

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q2.sh

Enter the number u want to check the prime and its palindrome prime property for: 29

29 is prime

its palindrome is not prime

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q2.sh

Enter the number u want to check the prime and its palindrome prime property for: 11

11 is prime

its palindrome is also prime

nida@nida-PC MINGW64 ~/TerminalCommnads/stage3/func

$ ./q2.sh

Enter the number u want to check the prime and its palindrome prime property for: 101

101 is prime

its palindrome is also prime