**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

#!/bin/bash -x

degree(){

echo "1. Convert Celsius temperature into Fahrenheit"

echo "2. Convert Fahrenheit temperatures into Celsius"

echo -n "Select your choice (1-2) : "

read choice

if [ $choice -eq 1 ]

then

echo -n "Enter celsius temperature (C) : "

read celsiustemperature

celsius=`awk -v n= "BEGIN {print ($celsiustemperature - 32) \* (5/9)}"`

echo $celsius

elif [ $choice -eq 2 ]

then

echo -n "Enter temperature (F) : "

read fahrenheittemperature

fahrenheit=`awk -v n= "BEGIN {print (9/5) \* ($fahrenheittemperature + 32)}"`

echo $fahrenheit

else

echo "Please select 1 or 2 only"

exit 1

fi

}

degree

echo $?

Output:

AKHIL@DESKTOP-43E18QK MINGW64 ~

$ ./function1ex2.sh

+ degree

+ echo '\*\*\* Converting between the different temperature scales \*\*\*'

\*\*\* Converting between the different temperature scales \*\*\*

+ echo '1. Convert Celsius temperature into Fahrenheit'

1. Convert Celsius temperature into Fahrenheit

+ echo '2. Convert Fahrenheit temperatures into Celsius'

2. Convert Fahrenheit temperatures into Celsius

+ echo -n 'Select your choice (1-2) : '

Select your choice (1-2) : + read choice

2

+ '[' 2 -eq 1 ']'

+ '[' 2 -eq 2 ']'

+ echo -n 'Enter temperature (F) : '

Enter temperature (F) : + read fahrenheittemperature

32

++ awk -v n= 'BEGIN {print (9/5) \* (32 + 32)}'

+ fahrenheit=115.2

+ echo 115.2

115.2

+ echo 0

0

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

#! /bin/bash -x

echo "enter a number"

read n

number=$n

reverse=0

palindrome(){

while [ $n -gt 0 ]

do

a=`expr $n % 10`

n=`expr $n / 10`

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

done

}

palindrome

echo $reverse

if [ $number -eq $reverse ]

then

echo "number is a palindrome"

else

echo "number is not a palindrome"

fi

output:

AKHIL@DESKTOP-43E18QK MINGW64 ~

$ ./polindrome.sh

+ echo 'enter a number'

enter a number

+ read n

212

+ number=212

+ reverse=0

+ palindrome

+ '[' 212 -gt 0 ']'

++ expr 212 % 10

+ a=2

++ expr 212 / 10

+ n=21

++ expr 0 '\*' 10 + 2

+ reverse=2

+ '[' 21 -gt 0 ']'

++ expr 21 % 10

+ a=1

++ expr 21 / 10

+ n=2

++ expr 2 '\*' 10 + 1

+ reverse=21

+ '[' 2 -gt 0 ']'

++ expr 2 % 10

+ a=2

++ expr 2 / 10

+ n=0

++ expr 21 '\*' 10 + 2

+ reverse=212

+ '[' 0 -gt 0 ']'

+ echo 212

212

+ '[' 212 -eq 212 ']'

+ echo 'number is a palindrome'

number is a 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**

#!/bin/bash -x

prime(){

echo "enter a num"

read n

number=$n

reverse=0

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

do

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

then

echo "$n is not a prime number"

exit

fi

done

echo "$n is a prime number"

}

palindrome(){

while [ $n -gt 0 ]

do

a=`expr $n % 10`

n=`expr $n / 10`

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

done

echo $reverse

if [ $number == $reverse ]

then

echo "number is a palindrome"

else

echo "number is not a palindrome"

fi

}

prime

palindrome

exit $?

echo $number number is palindrome and prime

output:

AKHIL@DESKTOP-43E18QK MINGW64 ~

$ ./primeandpolindrome.sh

+ prime

+ echo 'enter a num'

enter a num

+ read n

11

+ number=11

+ reverse=0

+ (( i=2 ))

+ (( i<=n/2 ))

+ '[' 1 -eq 0 ']'

+ (( i++ ))

+ (( i<=n/2 ))

+ '[' 2 -eq 0 ']'

+ (( i++ ))

+ (( i<=n/2 ))

+ '[' 3 -eq 0 ']'

+ (( i++ ))

+ (( i<=n/2 ))

+ '[' 1 -eq 0 ']'

+ (( i++ ))

+ (( i<=n/2 ))

+ echo '11 is a prime number'

11 is a prime number

+ palindrome

+ '[' 11 -gt 0 ']'

++ expr 11 % 10

+ a=1

++ expr 11 / 10

+ n=1

++ expr 0 '\*' 10 + 1

+ reverse=1

+ '[' 1 -gt 0 ']'

++ expr 1 % 10

+ a=1

++ expr 1 / 10

+ n=0

++ expr 1 '\*' 10 + 1

+ reverse=11

+ '[' 0 -gt 0 ']'

+ echo 11

11

+ '[' 11 == 11 ']'

+ echo 'number is a palindrome'

number is a palindrome

+ exit 0