Practice Problems DAY 6 PROB 1

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.

```bash
#!/bin/bash


echo "Enter the n value"
read n

p=1

for((i=0; i<$n; ++i))
do
        if(($i==0))
        then
                p=1
        else
                p=$(($p*2))
        fi

        echo $p

done
```

---

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

```bash
#!/bin/bash

harmonic=1

read -p "Enter the Nth number" n

#harmonic=$(($harmonic + 1))

#echo $harmonic
```

```
for (( i=2; i<=$n; i++))
do
  res=`echo "scale=2; 1 / $i" | bc`
  harmonic=`echo "scale=2; $harmonic + $res" | bc`
  #harmonic=$(($harmonic + $res))



done


echo $harmonic
```

---

3)Write a program that takes a input and determines if the number is a prime.
```
echo "Please Enter the number"
read number
i=2
f=0

for ((i=2; i<=number; i++))
do
while test $i -le `expr $number / 2`
do


if test `expr $number % $i` -eq 0
then
f=1
fi


i=`expr $i + 1`
done
if test $f -eq 1
then
echo "Not Prime"
else
echo "Prime"
fi
done
```

---

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

```
read -p "Enter start number and end range to find prime number" num1
read -p "Enter start number and end range to find prime number" num2
```

```
count=0

if((num2 <2))
then
        echo "There are no prime numbers"



else
        echo "Prime numbers in the given range are"

        temp=$num1

        if(($num1 % 2 ==0))
        then
             num1=`expr $num1 + 1`
        fi


                for((i=$num1; i<=$num2; i=$i+2))
                do
                        flag=0
                        for((j=2; j<=$i/2; j++))
                        do
                                u=$(($i % $j))
                                if(($u == 0))
                                then
                                        flag=1
                                        break;
                                fi

                        done

                        if((flag == 0))
                        then
                                echo $i
                        fi
                done
fi
```

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

```
echo "Enter a number"
read num

fact=1
```

```
while [ $num -gt 1 ]
do
  fact=$((fact * num))  #fact = fact * num
  num=$((num - 1))      #num = num - 1
done

echo $fact
```

---

6) Write a program to compute Factors of a number N using prime factorization method.

```
#!/bin/bash


read -p "Enter the number" n

while ((n%2 ==0))
do

        n=$n/2
done

a=$(bc <<< "scale=0; sqrt($n)")
for((i=3; i<=$a; i= i+2))
do
        while(($n %$i == 0))

        do
                echo $i
                n=$n/$i
        done
done

if(($n > 2))
then
        echo $n
fi
```

Practice Problems DAY 6 PROB 2

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..

```
#!/bin/bash


echo "Enter the n value"
read n

p=1

for((i=0; i<$n; ++i))
do
    if(($i==0))
    then
        p=1
    else
        p=$(($p*2))
    fi

    echo $p

done
```

---

2)

---

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

```
#!/bin/bash

count=0
count1=0

while(( $count <= 11 ||$count1 <=11))
do
  FLIP=$(($RANDOM %2))
```

```
   if [ $FLIP -eq 1 ];then
      echo "Heads"
      count1=`expr $count1 + 1`

   else
      echo "tails"
      count=`expr $count + 1`
   fi

   if(($count ==11 || $count1==11))
   then
     if(($count1 ==11))
     then

        echo "Head won $count1 times and Tail won $count times"
     else

        echo "Tail won $count times and Head won $count1 times"
     fi
     break
   fi

done
```

PRACTICE PROBLEM DAY 6 PROB 3

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

```bash
#!/bin/bash


temprature(){

if [ $1 -eq 1 ]
then
   echo -n "Enter temperature in Celsius : "
   read tc

   tf=$(echo "scale=2;((9/5) * $tc) + 32" |bc)
   echo "$tc Celsius = $tf Fahrenheit"
elif [ $1 -eq 2 ]
then
   echo -n "Enter temperature in Fahrenheit : "
   read tf

   tc=$(echo "scale=2;(5/9)*($tf-32)"|bc)
   echo "$tf Fahrenheit = $tc Celsius"
else
   echo "Select the correct option please"
   exit 1
fi


}


echo "Temprature Conversion System"
echo "1. Convert Celsius temperature into Fahrenheit"
echo "2. Convert Fahrenheit temperatures into Celsius"
echo -n "Select your choice (1-2) : "
read choice

temprature $choice
```

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

```bash
#!/bin/bash
```

```bash
palindrome()
{

number=$1
no=$1

reverse=0

while(( $no > 0))
do
    a=`expr $no % 10`
    n=`expr $no / 10`
    reverse=`expr $reverse \* 10 + $a`

done
echo $reverse

if[ $number -eq $reverse ]
then
    echo "It's a palindrome"
else
    echo "It's not palindrome"

fi
}


read -p "Enter values" n
palindrome $n
```

---

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

```bash
#!/bin/bash

prime()
```

```
{
        numo=$1
        for((i=2; i<=numo/2;i++))
        do
                if(($((numo%i))== 0))
                then
                        echo "$num1 is not a prime number"
                        exit
                fi

        done

         echo "$num1 is a prime number"
}




palindrome(){
        n=$1

        number=$1
        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
        prime $reverse
}


read -p "Enter a number" num

palindrome $num
```