Forloop Whileloop functions practice problems

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.

ans)#!/bin/bash
```
echo "enter n value"
read n
pow=1
a=0
for (( i=1; i<=$n; i++ ))
do
if (( $n%10-gt$a ))
then
   pow=$(( 2*pow ))
    echo "power of 2 "$pow
else
    echo "invalid"
fi
done
```

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

ans)#!/bin/bash -x
```
echo Enter a number
read n
sum=0
for (( i=1; i<=$n; i++ ))
do
sum= $(( $sum + ( 10000 / $i ) ))
echo "Sum n series is" $sum
done
for (( i=1; i<=5; i++ ))
do
a=echo $sum  $i
echo -e "$a\\c\"
if [ $i -eq 1 ]
then
echo  "hello"
fi
done
```

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

ans#!/bin/bash -x

```
echo "enter any number"
read num
count=0

    for (( i=2; i<=$num; i++ ))
     do
     if [ $(( $num%$i )) -eq 0 ]
     then
        count=$(( $count + 1 ))
     fi
    done
    if  [[ $count -eq 2 ]]
     then
     echo "given number is Prime"
     else
     echo "not Prime"
     fi
```

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

ans)
```
#!/bin/bash -x
echo "enter start value"
read  num
echo  "give range"
read   final
for (( n=num ; n<=$final; n++ ))
do
f=0
    for (( m=1;m<=n; m++ ))
     do

      if ( $n%$m-eq 0 )
then

      f=$(( $f+1 ))
fi
     done
     if ( $f -eq 2 )
     then
```

```
    echo " Prime"
    else
    echo "not Prime"
fi
done
```

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

ans)
```
#!/bin/bash -x
echo "enter any number to find factorial "
read num
a=0
factorial=1
for (( counter=$num;  $counter-gt$a; counter-- ))
do
  factorial=$(( $factorial * $counter ))
done
echo $factorial
```

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

ans)
```
#!/bin/bash
echo "enter any number "
read  num
if [ $num -lt 2 ]
then
    echo "invalid entry"
else
     counter=0
      while [ $(( $num%2 )) -eq 0 ]
      do
       # prime[(( counter++ ))]=2
         num=$(( $num/2 ))
        done

      for (( i=3; $(($i*$i))<=num; ((i+=2)) ))
      do
        while [ $(( $num%$i )) -eq 0 ]
        do
```

```
        # prime[(( counetr++ ))]=$i
         num=$(( $num/$i ))
        done
       done
       echo $num
fi
```

7)**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**

ans)
```
#!/bin/bash -x
echo "enter n value"
read n
pow=1
a=256
i=1;
while [ $i -lt $n ]
do
if (( ( $n-le$a ) || ( $pow-eq$a ) ))
then

   pow=$(( 2*pow ))
    echo "power of 2 "$pow
    i=$(($i+1))

else
    echo "invalid"
fi
done
```

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

ans)
```
#!/bin/bash

printf "Choose (h)eads or (t)ails: "
read user_choice
count=1
while [ $count -lt 11 ];
do
if [ $user_choice != h ] && [ $user_choice != t ]; then
  echo "Invalid choice. Defaulting to (h)eads."
  user_choice=h
fi
```

```
computer_choice=$((RANDOM % 2 + 1))

if [ $computer_choice -eq 1 ]; then
  echo "Computer chose heads."
else
  echo "Computer chose tails."
fi

if [ $computer_choice -eq 1 ] && [ $user_choice = h ]; then
     echo "You win!"

        count=$(( $count+1 ))
        echo $count
elif [ $computer_choice -eq 1 ] && [ $user_choice = t  ]; then
   echo "You lose!"
elif [ $computer_choice -eq 2 ] && [ $user_choice = t ]; then
    echo "You win!"

     count=$(( $count+1 ))
     echo "$count"
else
  echo "You lose!"
fi

done
```

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

ans)

```
#!/bin/bash -x
echo "enter first number "
read num
echo "enter second number"
read num2
rev=0
rmdr=0

     function polindrome()
      {
         number=$num
         while [[ $number -gt 0 ]]
         do
            rmdr=$(( $number%10 ))
            number=$(( $number/10 ))
            rev=$(( $rev*10+$rmdr ))
         done
```

```
        if [[ $number -eq $rev ]]
        then
            echo "given number is palindrome"
         else
           echo "not polindrome"
       fi
    }
polindrome $num
```

10)**Take a number from user and check if the number is a Prime then show that its palindrome is also prime**

Ans)

```
#!/bin/bash
echo "enter any number"
read num

function myfunprime()
{

     i=2
    flag=0
   while [ $i-le$number / 2 ];
   do
   if [ $number % $i -eq 0 ];
   then
   flag=1
   fi

   i=$(( $i + 1 ))
  done
   if [ $flag -eq 1 ];
  then
  echo "The number is Not Prime"
   else
  echo "The number is Prime"
  fi

}
function myfunpolindrome()
{
   num=545
   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
}

resultprime=$( myfunprime )
resultpolindrome=$( myfunpolindrome )
if [ $resultprime -eq $resultpolindrome]
then
  echo "polindrome is also prime"
else
   echo "polindrome is not prime"
fi
```

11)**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)**

ans) step1

User want to select which conversion should be done whether fahrenheit to celsius or vise versa

Step2

```
Case $degree in
    1) Based on selection if he choice 1
       tf=$(echo "scale=2;((9/5) * $tc) + 32" |bc)
    2)based on selection if he choice 2
    tc=$(echo "scale=2;(5/9)*($tf-32)"|bc)
esac
```

12)Find the Magic Number

ans)

```
#!/bin/bash
echo "think any number between 1 to 100"
low=1
high=100;
```

```
half=50
while [ $low -le $high ]
do

       echo "enter 1 if there any number below   $half"
       echo "enter 2 if ther any number above  $half"
        echo "enter 3 if their is any number equal to $half"
       read num
      case $num in
      1)
            high=$(( $half-1 ))
        ;;
      2)      low=$(( $half+1 ))
        ;;
      3)echo "your number is $half"
            exit
        ;;
      *) echo "invalid number"
             exit
        ;;
        esac
       half=$(( ( $low+$high)/2 ))
Done
```

13)**Write a Program where a gambler**

ans#

```
#!/bin/bash
win=1
loss=0
wincount=0
losscount=0
money=100
goal1=200
goal2=0

while [ $goal-ne$money ] || [ $goal2-ne$money ]
do
        bet=$(( RANDOM % 2 ))
        if [ $bet-eq$win ]
         then
              echo "Bet wins"
               wincount=$(( $wincount + 1 ))
              money=$(( $money+1 ))
              echo "available money" $money
             if [ $money -eq $goal1 ]
```

```
            then
                echo "wincount"$wincount
                exit
            fi

            elif [ $bet-eq$loss ]
            then
                echo "bet loss"
                losscount=$(( $losscount + 1 ))
                money=$(( $money-1 ))
                echo  "available" $money
                if [ $money -eq 0 ]
                then
                    echo "losscount" $losscount
                    exit
                fi
            else
                echo exit
            fi
done
```