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

```
  GNU nano 6.4                                  6_3-1.sh
echo "Select the conversion:"
echo "F. Celsius to Fahrenheit"
echo "C. Fahrenheit to Celsius"
read -p "Enter your choice (F or C): " choice

case $choice in
  F|f)
    read -p "Enter temperature in Celsius (0 to 100): " degC
    if [ $degC -ge 0 ] && [ $degC -le 100 ]; then
      degF=$(( (degC * 9 / 5) + 32 ))
      echo "$degC°C is equal to $degF°F"
    else
      echo "Invalid temperature. Please enter a value between 0 and 100."
    fi
    ;;
  C|c)
    read -p "Enter temperature in Fahrenheit (32 to 212): " degF
    if [ $degF -ge 32 ] && [ $degF -le 212 ]; then
      degC=$(( (degF - 32) * 5 / 9 ))
      echo "$degF°F is equal to $degC°C"
    else
      echo "Invalid temperature. Please enter a value between 32 and 212."
    fi
    ;;
  *)
    echo "Invalid choice. Please enter F or C."
    ;;
esac
```

```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-1.sh
Select the conversion:
F. Celsius to Fahrenheit
C. Fahrenheit to Celsius
Enter your choice (F or C): C
Enter temperature in Fahrenheit (32 to 212): 54
54°F is equal to 12°C

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-1.sh
Select the conversion:
F. Celsius to Fahrenheit
C. Fahrenheit to Celsius
Enter your choice (F or C): f
Enter temperature in Celsius (0 to 100): 23
23°C is equal to 73°F
```

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

```
  GNU nano 6.4                                    6_3-2.sh
function is_palindrome() {
  number=$1
  reverse=0
  temp=$number

  while [ $temp -gt 0 ]; do
    last_digit=$((temp % 10))
    reverse=$((reverse * 10 + last_digit))
    temp=$((temp / 10))
  done

  if [ $number -eq $reverse ]; then
    echo "$number is a palindrome"
  else
    echo "$number is not a palindrome"
  fi
}

echo "Enter a number:"
read num
is_palindrome $num
```

```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-2.sh
Enter a number:
343
343 is a palindrome

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-2.sh
Enter a number:
5243425
5243425 is a palindrome

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-2.sh
Enter a number:
253
253 is not a palindrome

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-2.sh
Enter a number:
0
0 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

```
  GNU nano 6.4                                  6_3-3.sh
is_prime() {
  num=$1
  if [ $num -le 1 ]; then
    return 1
  fi
  for ((i=2; i*i<=num; i++)); do
    if [ $((num % i)) -eq 0 ]; then
      return 1
    fi
  done
  return 0
}

get_palindrome() {
  num=$1
  palindrome=0
  while [ $num -ne 0 ]; do
    remainder=$((num % 10))
    palindrome=$((palindrome * 10 + remainder))
    num=$((num / 10))
  done
  echo $palindrome
}

read -p "Enter a number: " number

is_prime $number
prime_status=$?

if [ $prime_status -eq 0 ]; then
  echo "$number is prime."
  palindrome=$(get_palindrome $number)
  echo "Its palindrome is $palindrome."

  is_prime $palindrome
  palindrome_prime_status=$?

  if [ $palindrome_prime_status -eq 0 ]; then
    echo "The palindrome ($palindrome) is also prime."
  else
    echo "The palindrome ($palindrome) is not prime."
  fi
else
  echo "$number is not prime."
fi
```

```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-3.sh
Enter a number: 13
13 is prime.
Its palindrome is 31.
The palindrome (31) is also prime.

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-3.sh
Enter a number: 141
141 is not prime.

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D6 (main)
$ sh 6_3-3.sh
Enter a number: 51
51 is not prime.
```