**# 1 Practice simple functions**

def greet():

    print("Hello this is my greet function.")

    print("I have to print three statements.")

    print("This is the last one, bye madafaka.")

greet()

**# 2 Function with input**

def greet\_with\_input(name):

    print (f"Hello {name}.")

names = input ("What is you name? ")

greet\_with\_input (names)

**# 3 Function with more than one input**

def greet\_with\_inputs(name, location):

    print (f"Hey {name}, how are things going in {location}?")

names = input ("Hello, what's your name?")

locations = input ("Where are you? ")

greet\_with\_inputs (names, locations)

#also posible to assing the values manually

greet\_with\_inputs (location = "cocina", name = "Edurado")

**# 4 Paint are calculator**

def number\_of\_cans(height, width):

    coverage = 5

    cans = (height \* width)/coverage

    cans = round(cans)

    print (f"\n\nYou will need to buy {cans} cans to paint your wall.\n")

print ("Hello, here you can calculate how many cans you need to paint your wall.")

heights = int(input ("\nWhat is the height of your wall: "))

widths = int(input ("\nWhat is the width? "))

number\_of\_cans (heights, widths)

input ("Press the enter key to exit.")

**# 5 Prime number checker**

#function

def prime\_checker(n):

    n = int(n)

    test1 = n % 2

    test2 = n % 3

    test3 = n % 5

    test4 = n % 7

    if n != 2 and n != 3 and n != 5 and n != 7:

        if test1 != 0 and test2 !=0 and test3 != 0 and test4 != 0:

            print (f"{n} is a prime number.")

        else:

            print (f"{n} is not a prime number.")

    else:

        print (f"{n} is a prime number.")

#main

number = input ("What is the number? ")

prime\_checker(number)

**# 6 prime number checker**

#function

def prime\_checker(n):

    n = int(n)

    test1 = n % 2

    test2 = n % 3

    test3 = n % 5

    test4 = n % 7

    if n != 2 and n != 3 and n != 5 and n != 7:

        if test1 != 0 and test2 !=0 and test3 != 0 and test4 != 0:

            print (f"{n} is a prime number.")

        else:

            print (f"{n} is not a prime number.")

    else:

        print (f"{n} is a prime number.")

#main

number = input ("What is the number? ")

prime\_checker(number)

**# 6 Prime number checker teacher solution / NO FUNCIONA**

def prime\_checker(n):

    is\_prime = True

    for i in range (2,n):

        if (n % i) == 0:

            if is\_prime == True: # With this if condition it works

                is\_prime = False

    if is\_prime == True:

        print ("It's a prime number")

    elif is\_prime == False:

        print ("It's not a prime number.")

user = int(input("number: "))

prime\_checker(user)

**# test day 8**

from alphabet import alphabet\_list

# from caesar\_cipher import cipher

# from caesar\_cipher import decipher

from caesar\_cipher import function\_selection

# from caesar\_cipher\_teacher import encrypt

continuar = "yes"

while continuar == "yes":

    selection = input ("Do you want encryp or decryp? ")

    message = input("message? :")

    shift = input ("Shift? ")

    function\_selection (selection, message, shift)

    continuar = input ("Do you want to continue? Yes or Not ")

print ("Bye bye madafaka!")

alphabet\_list = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', ]

from alphabet import alphabet\_list

def cipher (message, shift):

    shift = int(shift)

    new\_char = ""

    new\_message = ""

    while shift > 26:

            shift = shift % 26

    for char in message:

        if char in alphabet\_list:

            for i in range (0, len(alphabet\_list)):

                if i + shift < 26:

                    if char == alphabet\_list[i]:

                        new\_char = alphabet\_list[i+shift]

                        new\_message += new\_char

                elif i + shift >= 26:

                    if char == alphabet\_list[i]:

                        new\_char = alphabet\_list[i+shift-26]

                        new\_message += new\_char

        else:

            new\_message += char

    new\_message = str(new\_message)

    print (f"The encoded text is {new\_message}")

def decipher (message, shift):

    shift = int(shift)

    new\_char = ""

    new\_message = ""

    while shift > 26:

            shift = shift % 26

    for char in message:

        if char in alphabet\_list:

            for i in range (0, len(alphabet\_list)):

                if i + shift >= 0:

                    if char == alphabet\_list[i]:

                        new\_char = alphabet\_list[i-shift]

                        new\_message += new\_char

                elif i + shift < 0:

                    if char == alphabet\_list[i]:

                        new\_char = alphabet\_list[i-shift+26]

                        new\_message += new\_char

        else:

            new\_message += char

    new\_message = str(new\_message)

    print (f"The encoded text is {new\_message}")

def function\_selection (select, mensaje, salto):

    if select == "encryp":

        cipher(mensaje, salto)

    elif select == "decryp":

        decipher(mensaje, salto)