

Programando con Python - Solución

Pregunta 1

Crea una función que reciba los tres coeficientes *a*, *b* y *c* para resolver una ecuación de segundo grado. Muestra la solución por pantalla y ayúdate de la librería *math* para acceder a la función raíz cuadrada.

Solución

```
import math
def eq2(a, b, c):
    return [(-b + math.sqrt(math.pow(b, 2) - 4 * a * c)) / (2 * a),
            (-b - math.sqrt(math.pow(b, 2) - 4 * a * c)) / (2 * a)]

eq2(1, 2, -1) # Resuelve la ecuación  $x^2 + 2x - 1 = 0$ 
```

```
[0.41421356237309515, -2.414213562373095]
```

Pregunta 2

Crea una función que lea una frase de teclado y nos diga si es o no un palíndromo (frase que se lee igual de izquierda a derecha o al revés como por ejemplo “La ruta nos aporó otro paso natural”).

Solución

```
def esPalindromo():
    s = input("Escribe una frase: ")
    l = [] # Creamos una lista vacía
    s = s.lower() # Pasamos todos los caracteres a minúscula
    for c in s:
        if c != " ": # Eliminamos los espacios en blanco
            l.append(c) # Añadimos en la lista todas las letras

    isPalindrome = True
    for i in range(int(len(l) / 2)):
        if l[i] != l[len(l) - (i + 1)]: # Comparamos si los pares de letras son iguales
            isPalindrome = False # Si algún par es diferente, la frase ya no es un palíndromo
            break
    return isPalindrome
```

Pregunta 3

Crea un diccionario que tenga por claves los números del 1 al 10 y como valores sus raíces cuadradas.

Solución

```
import math
d = {}
for i in range(1, 11):
    d[i] = math.sqrt(i)
```

Pregunta 4

Crea un diccionario que tenga como claves las letras del alfabeto castellano y como valores los símbolos del código morse (los tienes todos en la Wikipedia). A continuación crea una función que lea una frase del teclado y te la convierta a morse utilizando el diccionario anterior.

Solución

```
morse = {
    "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": "--.."
}

def toMorse():
    s = input("Introduce una frase: ")
    s = s.lower()
    message = ""
    for c in s:
        if c == " ":
            message += " "
            continue
        message += morse[c] + " "
    return message
```

Pregunta 5

Crea una función que dados dos diccionarios nos diga qué claves están presentes en ambos.

Solución

```
def clavesIguales(d1, d2):
    l = []
    for key in d1:
        if key in d2:
            l.append(key)
```

```

    return 1

d1 = {"pera": 1, "manzana": 2, "cereza": 5, "sandia": 3}
d2 = {"manzana": 5, "melon": 1, "pera": 7, "melocoton": 4}

clavesIguales(d1, d2)

['pera', 'manzana']

```

Pregunta 6

Crea una función que dado un número N nos diga si es primo o no (tiene que ir dividiendo por todos los números x comprendidos entre 2 y el número $N - 1$ y ver si la división de $\frac{N}{x}$ tiene resto cero o no).

Solución

```

def esPrimo(N):
    divisores = []

    for i in range(1, N):
        if N % i == 0:
            divisores.append(i)

    if len(divisores) > 1:
        return False
    return True

esPrimo(23)

```

True

Pregunta 7

Investiga la documentación de la clase `string` y crea un método que lea una frase del teclado y escriba la primera letra de cada palabra en Mayúscula.

Solución

```

def toTitle():
    s = input("Escribe una frase:")
    s = s.title()
    return s

```

Pregunta 8

Crea una función que calcule el máximo común divisor de dos números introducidos por el usuario por teclado.

Solución

```
def maximoComunDivisor(a, b):  
    aux = 0  
  
    while b != 0:  
        aux = b  
        b = a % b  
        a = aux  
  
    return a
```

Pregunta 9

Investiga el Cifrado del César y crea una función que lo reproduzca en **Python**. Cada letra del mensaje original se desplaza tres posiciones en el alfabeto estándar. La A se convierte en la D, la B se convierte en la E, la C se convierte en la F... y cuando se acaba el alfabeto se le vuelve a dar la vuelta: la X se convierte en la A, la Y en la B y la Z en la C. Los números no sufren ninguna modificación.

Solución

```
def cifradoCesar(letra):  
    letra = letra.upper()  
    cesar = 65 + (ord(letra) - 65 + 3) % 26  
    return chr(cesar)
```

Pregunta 10

Dado una lista de nombres de persona, escribe una función que los ordene de tres formas diferentes:

- A. De forma alfabética
- B. De forma alfabética invertida
- C. De nombre más corto al más largo.

Solución

```
nombres = ["Marc", "Aurelio", "Blanca", "Paz", "Carlitos", "Felicitas"]  
  
sorted(nombres)
```

```
['Aurelio', 'Blanca', 'Carlitos', 'Felicitas', 'Marc', 'Paz']
```

```
sorted(nombres, reverse = True)
```

```
['Paz', 'Marc', 'Felicitas', 'Carlitos', 'Blanca', 'Aurelio']
```

```
sorted(nombres, key = lambda x: len(x))
```

```
['Paz', 'Marc', 'Blanca', 'Aurelio', 'Carlitos', 'Felicitas']
```