Estructuras de datos en Python. Tarea 01

Alberto Simón

17/04/2025

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.

```
import math

def solve_second_grade(a, b, c):
    disc = b**2 - 4*a*c
    if disc < 0:
        print("No tiene soluciones reales")
    else if disc == 0:
        x = (-b) / (2*a)
        print("Solución única:", x)
    else
        x1 = (-b + math.sqrt(disc)) / (2*a)
        x2 = (-b - math.sqrt(disc)) / (2*a)
        print("Soluciones:", x1, x2)
```

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 aportó otro paso natural”).

```
def is_palindrome(sentence):
    sentence_clean = sentence(" ", "").lower()
    return sentence_clean == sentence_clean[::-1]
```

Pregunta 3

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

```
import math
dic = {x: math.sqrt(x) for x in range(1, 11)}
print(dic)
```

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.

```
morse = {
    'a': '.-', 'b': '-...', 'c': '-.-.', 'd': '-..', 'e': '.', 'f': '..-.', 'g': '--.', 'h': '....',
```

```
    'i': '..', 'j': '.---', 'k': '-.-', 'l': '.-..', 'm': '--', 'n': '-.', 'ñ': '--.--', 'o': '---',
    'p': '.--.', 'q': '--.-', 'r': '.-.', 's': '...', 't': '-', 'u': '..-', 'v': '...-', 'w': '.--',
    'x': '-..-', 'y': '-.--', 'z': '--..'
}

def to_morse(frase):
    sentence_clean = frase.lower()
    res = ""
    for l in sentence_clean:
        if l in morse:
            res += morse[l] + " "
        elif l == " ":
            res += "/ "
    print(res)
```

Pregunta 5

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

```
def keys_common(d1, d2):
    return list(d1.keys() & d2.keys())
```

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 N x tiene resto cero o no).

```
def is_prime(n):
    if n <= 2:
        return True
    for x in range(2, n):
        if n % x == 0:
            return False
    return 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.

```
def capitalize_sentence(sentence):
    return sentence.title()
```

Pregunta 8

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

```
import math

def mcd():
    a = int(input("Primer número: "))
    b = int(input("Segundo número: "))
    print("MCD:", math.gcd(a, b))
```

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 X en la C. Los números no sufren ninguna modificación.

```
def caesar_cypher(text):
    res = ""
    for c in text:
        if c.isalpha():
            base = ord('A') if c.isupper() else ord('a')
            cyphrchr = chr((ord(c) - base + 3) % 26 + base)
            res += cyphrchr
        else:
            res += c
    print(res)
```

Pregunta 10

Dado una lista de nombres de persona, escribe una función que los ordene de tres formas diferentes: - De forma alfabética - De forma alfabética invertida - De nombre más corto al más largo.

```
def names_sort(names_list):
    print("Alfabético:", sorted(names_list))
    print("Invertido:", sorted(names_list, reverse=True))
    print("Por longitud:", sorted(names_list, key=len))
```