

Parte A: El programa anexo descifra por fuerza bruta un texto cifrado utilizando cifrado César, analízelo y explique detalladamente su funcionamiento.

El programa consta de 6 partes principales.

5 funciones y 1 parte principal:

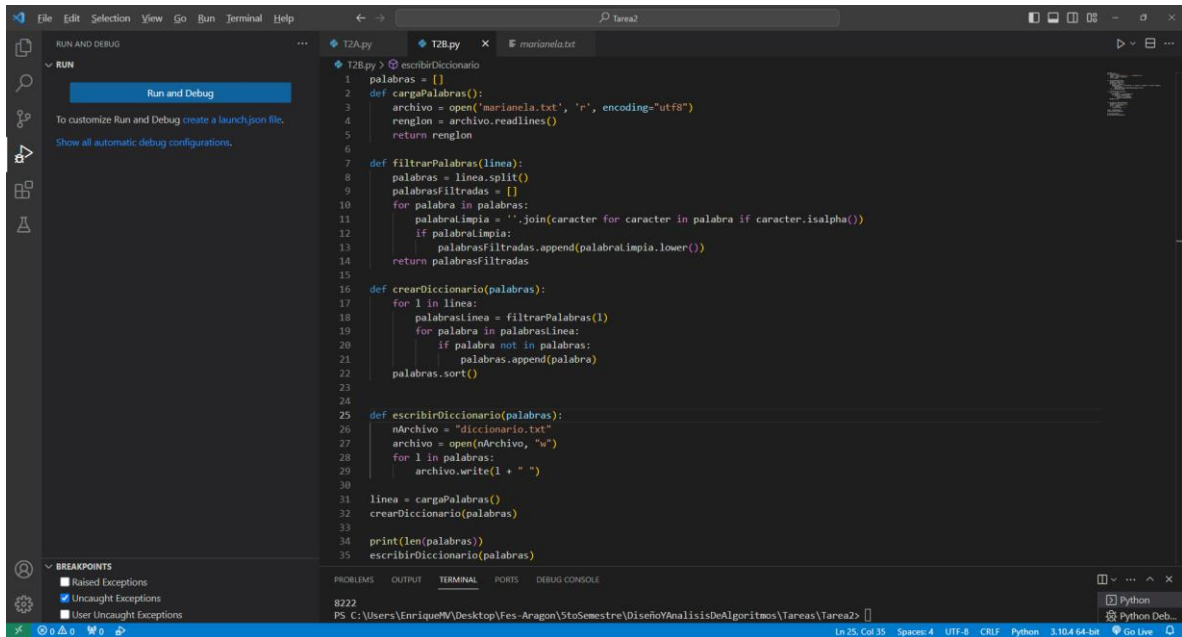
Funciones:

1. **cargaPalabras:** Esta función se encarga de obtener las palabras dentro del archivo “words” y las almacena en una variable en forma de diccionario, así como contabilizarlas.
2. **cargaCifrado:** Esta función se encarga de abrir el texto cifrado con Cesar del archivo “textoCifrado” y lo guarda en una variable.
3. **CifraCesar:** Esta función se encarga de descifrar el texto cifrado:
 - Pasa todo el texto cifrado a minúsculas
 - Crea una variable con las letras del alfabeto
 - Realiza 1 recorridos, se encarga de analizar carácter por carácter, recibe el carácter y lo compara con el alfabeto, obtiene su posición dentro del alfabeto, después, suma la posición a la llave y obtiene su modulo entre 26 para registra el nuevo carácter con una letra de la variable “alfabeto” en la posición resultante de la operación anterior, así hasta terminar ambos recorridos.
 - Si no hay un cambio en los caracteres, se agrega el mismo que se analizó
4. **descifrarCesar:** Manda a llamar la función CifraCesar.
5. **getAciertos:** Realiza la comparación con las palabras de la cadena y del diccionario, si coinciden, se aumenta un contador.

Parte principal:

- Crea variables que tendrán los datos de las funciones CargaPalabras y cargaCifrado
- Crea variables que contengan los datos de Máximo de aciertos y una posible llave
- Realiza un recorrido del 0-26 que analizará los datos obtenidos de las variables anteriores, descifrando la cadena y luego utilizando getAciertos, saber si las palabras obtenidas del descifrado son palabras reales del diccionario, si es así, termina el recorrido e imprime el resultado.

Parte B: Escriba un programa que cree un diccionario, y utilícelo para crear un diccionario en español, sólo minúsculas, sin signos de puntuación, etc, y guárdelo en un archivo con el mismo formato que tiene el diccionario en inglés.



```
1 palabras = []
2 def cargaPalabras():
3     archivo = open('marianela.txt', 'r', encoding="utf8")
4     renglon = archivo.readlines()
5     return renglon
6
7 def filtrarPalabras(linea):
8     palabras = linea.split()
9     palabrasFiltradas = []
10    for palabra in palabras:
11        palabraLimpia = ''.join(caracter for caracter in palabra if caracter.isalpha())
12        if palabraLimpia:
13            palabrasFiltradas.append(palabraLimpia.lower())
14    return palabrasFiltradas
15
16 def crearDiccionario(palabras):
17     for l in linea:
18         palabrasLinea = filtrarPalabras(l)
19         for palabra in palabrasLinea:
20             if palabra not in palabras:
21                 palabras.append(palabra)
22     palabras.sort()
23
24
25 def escribirDiccionario(palabras):
26     nArchivo = "diccionario.txt"
27     archivo = open(nArchivo, "w")
28     for l in palabras:
29         archivo.write(l + " ")
30
31 linea = cargaPalabras()
32 crearDiccionario(palabras)
33
34 print(len(palabras))
35 escribirDiccionario(palabras)
```

The screenshot shows a Python IDE with a dark theme. The main editor window displays a Python script named 'T2B.py'. The script defines functions to load words from a file, filter them to lowercase letters only, create a dictionary, and write it to a new file. The left sidebar shows the 'RUN AND DEBUG' panel with a 'Run and Debug' button. The bottom status bar indicates the file is at line 25, column 35, using UTF-8 encoding, and the Python version is 3.10.4 64-bit.