```
1 def crea_archivo(nombre):
2 file = open(nombre, "w")
3 for i in range(3):
4
      frase = input("Dame una frase: ")
5
      file.write(frase)
6
      file.write("\n")
7 file.close()
1 def limpia_archivo(nombre, nombreD):
2 file0 = open(nombre, "r")
    fileD = open(nombreD, "w+")
3
   c = file0.read(1)
    seguros = " abcdefghijklmnopqrstuvwxyzñáéíóú0123456789"
5
    while c != "":
6
7
      if c.lower() in seguros:
8
        fileD.write(c)
      elif c == "n":
9
       fileD.write(" ")
10
11
      c = file0.read(1)
12 fileO.close()
13 fileD.close()
14
1 def obtiene_palabras(nombre):
2 file = open(nombre, "r")
3 texto = file.read()
4 # print(texto)
5 lista = texto.split(" ")
6 for elemento in lista:
      if elemento == '':
7
        lista.remove('')
8
   return lista
1 def promedio_letras(lista):
2 \quad acum = 0
3 for palabra in lista:
      acum = acum + len(palabra)
4
5 return acum / len(lista)
1 def digitos_consonantes(lista):
2 d = 0
3
    c = 0
4 digitos = "0123456789"
5 consonantes = "bcdfghjklmnpqrstvwxyz"
6
   for palabra in lista:
7
      if palabra[0] in digitos:
8
        d = d + 1
9
      elif palabra[0].lower() in consonantes:
        c = c + 1
10
11 return (d, c)
1 def diccionario_abecedario(lista):
    abecedario = "abcdefghijklmnñopqrstuvwxyz"
    diccionario = {}
3
4
    for caracter in abecedario:
      listav = []
5
6
      for palabra in lista:
7
        if palabra[0].lower() == caracter:
8
          listav.append(palabra)
      diccionario[caracter] = listav
   return diccionario
```

```
1 def imprime_dicionario(diccionario):
 2 for key in diccionario:
       print(key, ":", diccionario[key])
 1 def cuenta_signos(nombre):
 2 file = open(nombre, "r")
 3 c = file.read(1)
 4 pa = 0
 5
    pc = 0
 6
    sa = 0
 7
    sc = 0
 8 while c != "":
 9
     if c == "(":
10
       pa = pa + 1
     elif c == ")":
11
       pc = pc + 1
12
13
     elif c == "¿":
14
       sa = sa + 1
     elif c == "?":
15
16
       sc = sc + 1
17
      c = file.read(1)
18
    print("Paréntesis abiertos: ", pa)
19
    print("Paréntesis cerrados: ", pc)
20 print("Signo interrogación abierto: ", sa)
21 print("Signo interrogación abierto: ", sc)
 1 def main():
    nombre = input("Dame el nombre del archivo: ")
 3
    nombreD = "copia.txt"
 4
    limpia_archivo(nombre, nombreD)
 5
    print("\nLISTA DE PALABRAS")
 6
    lista = obtiene_palabras(nombreD)
 7
    print(lista)
    print("\nPROMEDIO DE LETRAS POR PALABRA")
 8
 9
    res = promedio_letras(lista)
10
    print("El promedio de palabras es: %.2f" % res)
11 tupla = digitos_consonantes(lista)
12 print("\nCANTIDAD DE PALABRAS")
13 print(tupla)
14 print("La cantidad de palabras que comienzan con dígito son: ", tupla[0])
15 print("La cantidad de palabras que comienzan con consonantes son: ", tupla[1])
16  d = diccionario_abecedario(lista)
17
    print("\nDICCIONARIO")
18
    print(d)
19
    print("\nIMPRIME DICCIONARIO")
20
    imprime dicionario(d)
21 print("\nCUENTA SIGNOS")
22
    cuenta_signos(nombre)
23
24 main()
    Dame el nombre del archivo: ElAlquimista.txt
    LISTA DE PALABRAS
    ['Libro', 'El', 'Alquimista', 'El', 'Alquimista', 'es', 'un', 'libro', 'escrito', 'por', 'el', 'brasileño', 'Paulo', 'Coelho', 'y', 'pub
    PROMEDIO DE LETRAS POR PALABRA
    El promedio de palabras es: 4.70
    CANTIDAD DE PALABRAS
    (18, 1272)
    La cantidad de palabras que comienzan con dígito son: 18
    La cantidad de palabras que comienzan con consonantes son: 1272
   DTCCTONARTO
    {'a': ['Alquimista', 'Alquimista', 'año', 'autor', 'ayudar', 'a', 'alcanzar', 'Alma', 'a', 'a', 'Alquimista', 'a', 'a', 'análisis', 'Alq
    IMPRIME DICCIONARIO
    a : ['Alquimista', 'Alquimista', 'año', 'autor', 'ayudar', 'a', 'alcanzar', 'Alma', 'a', 'a', 'Alquimista', 'a', 'a', 'análisis', 'Alqui
    b : [ˈbrasileño', 'busca', 'busca', 'busca', 'básicas', 'buenas', 'busca', 'búsqueda', 'busca', 'búsqueda', 'brasileño', 'Brasileña', 'b
```

```
c: ['Coelho', 'como', 'Carl', 'colectivo', 'cual', 'cosechado', 'Conozcamos', 'cerca', 'caso', 'compra', 'comerciante', 'comida', 'cama
d: ['del', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'del', 'del', 'de', 'fla', 'fla', 'fla', 'fla', 'fatian', 'forma', 'forma', 'forma', 'forma', 'forma', 'foturo', 'fsase', 'fan', 'Frases', 'guan', 'gitana', 'gan', 'gan', 'gan', 'gan', 'hace', 'haca', 'hace', 'haca', 'haca', 'haca', 'haca', 'haca', 'haca', 'haca', 'haca', 'hace', 'hombres i ['inconsciente', 'inspirado', 'interprete', 'identidad', 'inicia', 'inicio', 'Islam', 'inglés', 'indicado', 'iglesia', 'invitar', 'i
]: ['Jung', 'jefe', 'justamente', 'Junto', 'los', 'la', 'Leyenda', 'libro', 'lenguas', 'lecciones', 'libro', 'las', 'lectura', 'libro',

": ['mundo', 'mirada', 'millones', 'mundo', 'más', 'mostraba', 'Mientras', 'Más', 'mujer', 'monasterio', 'más', 'manera', 'metas', 'Mel

": ['mundo', 'mirada', 'millones', 'mundo', 'más', 'mostraba', 'Mientras', 'Más', 'mujer', 'monasterio', 'más', 'manera', 'metas', 'Mel

": ['osor', 'novela', 'no', 'no', 'no', 'no', 'norarativa', 'nos', 'nustros', 'nostros', 'nostros', 'no', 'obstáculos', 'nostros', 'no', 'observando', 'oasis', 'oasis', 'oarnativa', 'nos', 'nostros', 'nostros', 'no', 'obstáculos', 'ocasiones', 'ocurre', 'otro', 'obs

": ['obra', 'o', 'publicado', 'puede', 'psicólogo', 'pastor', 'Personal', 'personal', 'persigan', 'por', 'publicado', 'pue', 'que', 'que', 'que', 'que', '
```

Parentesis ablertos: 3
Paréntesis cerrados: 3
Signo interrogación abierto: 4
Signo interrogación abierto: 4

1