



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**



FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

DISEÑO Y ANÁLISIS DE ALGORITMOS

PROYECTO

DESCIFRAR TEXTO DE CIFRADO POR SUSTITUCIÓN

EQUIPO 17:

**MANZANO PONCE JOSUÉ
OLVERA MARTÍNEZ LEONARDO
ORTEGA BATÚN LUIS FERNANDO**

**PROFESOR:
PEREZ MEDEL MARCELO**

GRUPO: 1507

**FECHA: 23 DE NOVIEMBRE DEL 2023
SEMESTRE 2024-I**

TEXTO 1:

Texto cifrado:

Oitlt xlt onb ovjtc bd jtbjqt nib ngqq otqq vbr oixo vbr zxmmbo yxkt x egddtltmzt gm oigc nblqe: oibct nib xlt xdlxge ob olv xme oibct nib xlt xdlxge vbr ngqq crzztte Crzztcc gc qgkgmu vbrlctqd, qgkgmu nixo vbr eb, xme qgkgmu ibn vbr eb go Nitmtftl vbr dgme vbrlctqd bm oit cget bd oit yxwblgov, go gc ogyt ob jxrct xme ltdqtzo Kttj vbrl dxzt xqnxvc obnxle oit crmcigmt xme cixebnc ngqq dxqq htigme vbr Gd vbr ixft tftlvoigmu rmetl zbmolbq, vbr xlt mbo ybfgmu dxco tmbrui Go gc erlgmu brl exlktco ybytmoc oixo nt yrco dbzrc ob ctt oit qguio

(idioma detectado: Ingles)

Código:

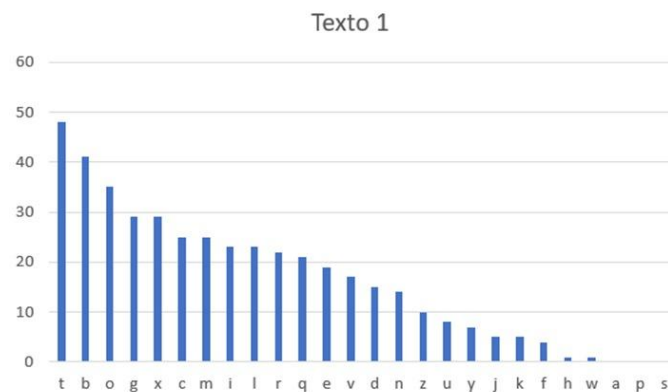
```
20 # Abre el archivo con el diccionario en ingles
21 def cargaPalabras():
22     archivo = open(os.path.join(ruta_carpeta, 'Ingles', 'ingles_diccionario.txt'), 'r')
23     renglon = archivo.readline()
24     palabras = renglon.split()
25     print (len(palabras), 'palabras leidas')
26     return palabras
27
28 # Abre el archivo donde viene el texto cifrado
29 def cargaCifrado():
30     archivo = open(os.path.join(ruta_carpeta, 'Texto1.txt'), 'r')
31     renglon = archivo.readline()
32     return renglon
33
34 def descifraSustituye(cadena, alfabetoLlave):
35     cadena = cadena.lower()
36     alfabeto = 'abcdefghijklmnopqrstuvwxyz'
37     nuevaCadena = ""
38     for letra in cadena: # recorre texto
39         if letra in alfabeto: # busca la letra en el alfabeto
40             posicion = alfabeto.find(letra) # obtiene posicion del alfabeto
41             nuevaCadena = nuevaCadena + alfabetoLlave[posicion] # pos nuevo alf
42         else:
43             nuevaCadena = nuevaCadena + letra # si no esta en alfabeto
44     return nuevaCadena
45
46 dic = cargaPalabras()
47 cifrado = cargaCifrado()
```

La primera función (cargaPalabras) abre el documento en donde guardamos nuestro diccionario de inglés y almacena estas palabras en una variable. La segunda función (cargaCifrado) obtiene el contenido del txt "Texto 1" en donde almacenamos nuestro primer texto cifrado. La última función (descifraSustituye) toma una cadena, que será nuestro texto a descifrar, y una llave, la cual hay que encontrar, y recorrerá la cadena carácter por

caracter obteniendo la posición de este en el alfabeto normal y sustituyendola por la posición en nuestra llave.

La salida del histograma del Texto cifrado 1, es el siguiente:

{'a': 0, 'b': 41, 'c': 25, 'd': 15, 'e': 19, 'f': 4, 'g': 29, 'h': 1, 'i': 23, 'j': 5, 'k': 5, 'l': 23, 'm': 25, 'n': 14, 'o': 35, 'p': 0, 'q': 21, 'r': 22, 's': 0, 't': 48, 'u': 8, 'v': 17, 'w': 1, 'x': 29, 'y': 7, 'z': 10}



En el histograma de frecuencias podemos ver que las letras que más se repiten son t, b, o, g lo que nos puede servir para la búsqueda del alfabeto, ya que será imposible encontrarles un valor, así que en nuestra llave la sustituiremos por una X.

```
posA="abcdefghijklmnopqrstuvwxyz"
posB="abcdefghijklmnopqrstuvwxyz"
posC="abcdefghijklmnopqrstuvwxyz"
posD="abcdefghijklmnopqrstuvwxyz"
posE="abcdefghijklmnopqrstuvwxyz"
posF="abcdefghijklmnopqrstuvwxyz"
posG="abcdefghijklmnopqrstuvwxyz"
posH="abcdefghijklmnopqrstuvwxyz"
posJ="abcdefghijklmnopqrstuvwxyz"
posK="abcdefghijklmnopqrstuvwxyz"
posL="abcdefghijklmnopqrstuvwxyz"
posM="abcdefghijklmnopqrstuvwxyz"
posN="abcdefghijklmnopqrstuvwxyz"
posO="abcdefghijklmnopqrstuvwxyz"
posP="abcdefghijklmnopqrstuvwxyz"
posQ="abcdefghijklmnopqrstuvwxyz"
posR="abcdefghijklmnopqrstuvwxyz"
posU="abcdefghijklmnopqrstuvwxyz"
posV="abcdefghijklmnopqrstuvwxyz"
posW="abcdefghijklmnopqrstuvwxyz"
posZ="abcdefghijklmnopqrstuvwxyz"
```

Empezando a descifrar por sustitución:

El problema al cual nos enfrentamos, si quisiéramos resolverlo por fuerza bruta, sería de un orden $n!$ lo cual representa un gran problema, así que para evitar tardar milenios en resolverlo, haremos una sustitución buscando las palabras que se puedan parecer que tengamos en nuestro diccionario, eliminando posibles letras del alfabeto hasta dar con su valor real (convirtiendo un problema np a p).

Lo que haremos será intentar buscar alguna palabra que tenga alguna característica en particular, letras que se repitan, una longitud determinada; sin combinamos estos dos elementos y buscamos dentro de nuestra amplia gama de palabras en el diccionario alguna que se acople con nuestras necesidades podremos encontrar algunas palabras candidatas para nuestra palabra cifrada. Si encontramos varias palabras podemos suponer que los primeros caracteres de nuestras palabras encontradas en el diccionario son buenos candidatos para el posible carácter cifrado, esto debemos hacerlo con cada carácter de la palabra descifrada, en caso de solo encontrar una palabra es más que excelente, ya los caracteres de esta palabra son los valores reales de los caracteres de nuestra palabra cifrada.

Esto hay que tomarlo en cuenta ya que hay que estar actualizando constantemente los valores de nuestros posibles caracteres, es posible que con la primer palabra obtengamos muchas palabras como candidatos, pero si continuamos actualizando los valores de nuestros posibles caracteres y volvemos a probar esta primer palabra, puede que nos de menos palabras de las que nos dio en un principio.

Esto hay que hacerlo constantemente y es probable que tengamos problemas, ya que hay una posibilidad que una palabra cifrada no se encuentre en nuestro diccionario, supongamos que una palabra hay dos palabras candidatas, solo una es la real, pero esa no se encuentra en nuestro diccionario, es entonces que nuestro código nos muestra solo una palabra y nos podemos quedar con la idea que esa palabra es la real, cuando no es así, esto es una situación complicada ya que podemos quedarnos un valor erróneo en un posible carácter y a medida que avancemos en nuestro código tendremos problemas para encontrar posibles palabras candidatas. Podemos resolver estos problemas o darnos cuenta dónde está el error intercambiando los caracteres que hemos descubierto en las palabras que están cifradas y el resultado meterlo en un traductor, en caso que el traductor no pueda identificar la palabra, es porque algún carácter tiene algún error.

Como se muestra a continuación, este fue el proceso que seguimos para descifrar el primer texto y básicamente es el mismo procedimiento que se explicó anteriormente, esto para ambos textos cifrados, el primero y el segundo.

Probando con: **Crzztcc**

```
50 posA = 'abcdefghijklmnopqrstuvwxyz' # 0
51 posB = 'abcdefghijklmnopqrstuvwxyz'
52 posC = 'abcdefghijklmnopqrstuvwxyz'
53 posD = 'abcdefghijklmnopqrstuvwxyz'
54 posE = 'abcdefghijklmnopqrstuvwxyz'
55 posF = 'abcdefghijklmnopqrstuvwxyz'
56 posG = 'abcdefghijklmnopqrstuvwxyz'
57 posH = 'abcdefghijklmnopqrstuvwxyz'
58 posI = 'abcdefghijklmnopqrstuvwxyz'
59 posJ = 'abcdefghijklmnopqrstuvwxyz'
60 posK = 'abcdefghijklmnopqrstuvwxyz'
61 posL = 'abcdefghijklmnopqrstuvwxyz'
62 posM = 'abcdefghijklmnopqrstuvwxyz'
63 posN = 'abcdefghijklmnopqrstuvwxyz'
64 posO = 'abcdefghijklmnopqrstuvwxyz'
65 posP = 'abcdefghijklmnopqrstuvwxyz' # 0
66 posQ = 'abcdefghijklmnopqrstuvwxyz'
67 posR = 'abcdefghijklmnopqrstuvwxyz'
68 posS = 'abcdefghijklmnopqrstuvwxyz' # 0
69 posT = 'abcdefghijklmnopqrstuvwxyz'
70 posU = 'abcdefghijklmnopqrstuvwxyz'
71 posV = 'abcdefghijklmnopqrstuvwxyz'
72 posW = 'abcdefghijklmnopqrstuvwxyz'
73 posX = 'abcdefghijklmnopqrstuvwxyz'
74 posY = 'abcdefghijklmnopqrstuvwxyz'
75 posZ = 'abcdefghijklmnopqrstuvwxyz'
76 c = 0
77
78 #probando con Crzztcc
79 for p in dic:
80     if len(p) == 7:
81         if p[0] == p[5] and p[0] == p[6] and p[2] == p[3]:
82             c = c + 1
83             print(p)
84 print(c)
```

Salida:

```
➡ Drive already mounted at /content/drive;
55900 palabras leidas
success
1
```

Probando con: **egddtlmzt**

```
88 posA = 'abcdefghijklmnopqrstuvwxyz'
89 posB = 'abcdefghijklmnopqrstuvwxyz'
90 posC = 's'
91 posD = 'abcdefghijklmnopqrstuvwxyz'
92 posE = 'abcdefghijklmnopqrstuvwxyz'
93 posF = 'abcdefghijklmnopqrstuvwxyz'
94 posG = 'abcdefghijklmnopqrstuvwxyz'
95 posH = 'abcdefghijklmnopqrstuvwxyz'
96 posI = 'abcdefghijklmnopqrstuvwxyz'
97 posJ = 'abcdefghijklmnopqrstuvwxyz'
98 posK = 'abcdefghijklmnopqrstuvwxyz'
99 posL = 'abcdefghijklmnopqrstuvwxyz'
100 posM = 'abcdefghijklmnopqrstuvwxyz'
101 posN = 'abcdefghijklmnopqrstuvwxyz'
102 posO = 'abcdefghijklmnopqrstuvwxyz'
103 posP = 'abcdefghijklmnopqrstuvwxyz'
104 posQ = 'abcdefghijklmnopqrstuvwxyz'
105 posR = 'u'
106 posS = 'abcdefghijklmnopqrstuvwxyz'
107 posT = 'e'
108 posU = 'abcdefghijklmnopqrstuvwxyz'
109 posV = 'abcdefghijklmnopqrstuvwxyz'
110 posW = 'abcdefghijklmnopqrstuvwxyz'
111 posX = 'abcdefghijklmnopqrstuvwxyz'
112 posY = 'abcdefghijklmnopqrstuvwxyz'
113 posZ = 'c'
114 c = 0
115
116 # probando con egddtlmzt
117 for p in dic:
118     if len(p) == 10:
119         if p[2] == p[3] and p[4] in posT and p[4] == p[6] and p[4] == p[9]:
120             c = c + 1
121             print(p)
122 print(c)
```

Salida:

```
Drive already mounted at /content/drive;
55900 palabras leidas
difference
1
```

Probando con: **zbmolbq**

```
126 posA = 'abhiijklmopqtvwxyz'
127 posB = 'abhiijklmopqtvwxyz'
128 posC = 's'
129 posD = 'f'
130 posE = 'd'
131 posF = 'abhiijklmopqtvwxyz'
132 posG = 'abhiijklmopqtvwxyz'
133 posH = 'abhiijklmopqtvwxyz'
134 posI = 'g' #error
135 posJ = 'abhiijklmopqtvwxyz'
136 posK = 'abhiijklmopqtvwxyz'
137 posL = 'r'
138 posM = 'n'
139 posN = 'abhiijklmopqtvwxyz'
140 posO = 'abhiijklmopqtvwxyz'
141 posP = 'abhiijklmopqtvwxyz'
142 posQ = 'abhiijklmopqtvwxyz'
143 posR = 'u'
144 posS = 'abhiijklmopqtvwxyz'
145 posT = 'e'
146 posU = 'abhiijklmopqtvwxyz'
147 posV = 'abhiijklmopqtvwxyz'
148 posW = 'abhiijklmopqtvwxyz'
149 posX = 'abhiijklmopqtvwxyz'
150 posY = 'abhiijklmopqtvwxyz'
151 posZ = 'c'
152 c = 0
153
154 # funcion para definir nuestro nuevo alfabeto quitando los caracteres encontrados
155 nuevoAlfabeto = 'abcdefghijklmnopqrstuvwxyz'
156 caracteresEncontrados = ['s', 'f', 'd', 'g', 'r', 'n', 'u', 'e', 'c']
157 for a in caracteresEncontrados:
158     nuevoAlfabeto = nuevoAlfabeto.replace(a, "")
159 print (nuevoAlfabeto)
160
161 # probando en zbmolbq
162 for p in dic:
163     if len(p) == 7:
164         if (p[0] in posZ and p[1] in posB and p[2] in posM and p[3] in posO
165             and p[4] in posL and p[1] == p[5] and p[6] in posQ):
166             c = c + 1
167             print(p)
168 print(c)
169 # Nos da 4 resultados así que usamos el abecedario reducido
170 # Al usar los abecedarios reducidos obtenemos un resultado: control
171
```

Salida:

```
Drive already mounted at
55900 palabras leidas
abhiijklmopqtvwxyz
control
1
```

Probando con: **ybytmoc**

```
172
173 posA = 'abhiijkmpqvwxzy'
174 posB = 'o'
175 posC = 's'
176 posD = 'f'
177 posE = 'd'
178 posF = 'abhiijkmpqvwxzy'
179 posG = 'abhiijkmpqvwxzy'
180 posH = 'abhiijkmpqvwxzy'
181 posI = 'g'
182 posJ = 'abhiijkmpqvwxzy'
183 posK = 'abhiijkmpqvwxzy'
184 posL = 'r'
185 posM = 'n'
186 posN = 'abhiijkmpqvwxzy'
187 posO = 't'
188 posP = 'abhiijkmpqvwxzy'
189 posQ = 'l'
190 posR = 'u'
191 posS = 'abhiijkmpqvwxzy'
192 posT = 'e'
193 posU = 'abhiijkmpqvwxzy'
194 posV = 'abhiijkmpqvwxzy'
195 posW = 'abhiijkmpqvwxzy'
196 posX = 'abhiijkmpqvwxzy'
197 posY = 'abhiijkmpqvwxzy'
198 posZ = 'c'
199 c = 0
200
201 # funcion para definir nuestro nuevo alfabeto quitando los caracteres encontrados
202 nuevoAlfabeto = 'abcdefghijklmnpqrstuvwxy'
203 caracteresEncontrados = ['o','s','f','d','g','r','n','t','l','u','e','c']
204 for a in caracteresEncontrados:
205     nuevoAlfabeto = nuevoAlfabeto.replace(a, "")
206 print (nuevoAlfabeto) # abhiijkmpqvwxzy
207
208 # probando en ybytmoc
209 for p in dic:
210     if len(p) == 7:
211         if (p[0] in posY and p[1] in posB and p[2] == p[0] and p[3] in posT
212             and p[4] in posM and p[5] in posO and p[6] in posC):
213             c = c + 1
214             print(p)
215 print(c)
216 # 1 resultado : moments
```

Salida:

```
Drive already mounted at /content/drive;
55900 palabras leidas
abhiijkmpqvwxzy
moments
1
```


Probando con: **Gd**

```
219 posA = 'abhijpgqvwxyz'
220 posB = 'o'
221 posC = 's'
222 posD = 'f'
223 posE = 'd'
224 posF = 'abhijpgqvwxyz'
225 posG = 'abhijpgqvwxyz'
226 posH = 'abhijpgqvwxyz'
227 posI = 'g'
228 posJ = 'abhijpgqvwxyz'
229 posK = 'abhijpgqvwxyz'
230 posL = 'r'
231 posM = 'n'
232 posN = 'abhijpgqvwxyz'
233 posO = 't'
234 posP = 'abhijpgqvwxyz'
235 posQ = 'l'
236 posR = 'u'
237 posS = 'abhijpgqvwxyz'
238 posT = 'e'
239 posU = 'abhijpgqvwxyz'
240 posV = 'abhijpgqvwxyz'
241 posW = 'abhijpgqvwxyz'
242 posX = 'abhijpgqvwxyz'
243 posY = 'm'
244 posZ = 'c'
245 c = 0
246
247 # funcion para definir nuestro nuevo alfabeto quitando los caracteres encontrados
248 nuevoAlfabeto = 'abcdefghijklmnopqrstuvwxyz'
249 caracteresEncontrados = ['o','s','f','d','g','r','n','t','l','u','e','m','c']
250 for a in caracteresEncontrados:
251     nuevoAlfabeto = nuevoAlfabeto.replace(a, "")
252 print (nuevoAlfabeto) # abhijpgqvwxyz
253
254 # probando en Gd
255 for p in dic:
256     if len(p) == 2:
257         if (p[0] in posG and p[1] in posD):
258             c = c + 1
259             print(p)
260 print(c)
261 # 1 resultado: if
```

Salida:

```
Drive already mounted at /content/drive;
55900 palabras leidas
abhijpgqvwxyz
if
1
```

Probando con: **yxwblgov**

[illegible]**Salida:**

```
Drive already mounted at /content/drive;
55900 palabras leidas
abhjklpqvwxyz
majority
1
!#<funcion para definir cuantos caracteres
```

Probando con: **bhkpqvwxyz**

```

312 posA = 'bhkpqvwxyz' # 0
313 posB = 'o'
314 posC = 's'
315 posD = 'f'
316 posE = 'd'
317 posF = 'bhkpqvwxyz'
318 posG = 'i'
319 posH = 'bhkpqvwxyz'
320 posI = 'g'
321 posJ = 'bhkpqvwxyz'
322 posK = 'bhkpqvwxyz'
323 posL = 'r'
324 posM = 'n'
325 posN = 'bhkpqvwxyz'
326 posO = 't'
327 posP = 'bhkpqvwxyz' # 0
328 posQ = 'l'
329 posR = 'u'
330 posS = 'bhkpqvwxyz' # 0
331 posT = 'e'
332 posU = 'bhkpqvwxyz'
333 posV = 'y'
334 posW = 'j'
335 posX = 'a'
336 posY = 'm'
337 posZ = 'c'
338 c = 0
339
340 # funcion para definir nuestro nuevo alfabeto quitando los caracteres encontrados
341 nuevoAlfabeto = 'abcdefghijklmnopqrstuvwxyz'
342 caracteresEncontrados = ['o','s','f','d','i','g','r','n','t','l','u','e','y',
343                           'j','a','m','c']
344 for a in caracteresEncontrados:
345     nuevoAlfabeto = nuevoAlfabeto.replace(a, "")
346 print (nuevoAlfabeto) # bhkpqvwxyz
347
348 # probando en xqnxvc
349 for p in dic:
350     if len(p) == 6:
351         if (p[0] in posX and p[1] in posQ and p[2] in posN and p[3] in posX
352             and p[4] in posV and p[5] in posC):
353             c = c + 1
354             print(p)
355 print(c)
356 # 1 resultado: always

```

Salida:

```
Drive already mounted at /content/drive;  
55900 palabras leidas  
bhkpgvwxz  
always  
1
```

Probando con: **jtbjqt**

[illegible]

Salida:

```
Drive already mounted at /content/drive; t
55900 palabras leidas
bhkpvxz
people
1
```

Probando con: **exlktco**

```

453 posA = 'bhqvzx' # 0
454 posB = 'o'
455 posC = 's'
456 posD = 'f'
457 posE = 'd'
458 posF = 'bhqvzx' # tentativa VZ -> tftlvoigmu | E_ERYTGIN_
459 posG = 'i'
460 posH = 'bhqvzx'
461 posI = 'h'
462 #tftlvoigmu | E_ERYTGIN_ (no concuerda con que i sea g
463 #asi que se revisa y se cometió un error en la primera combinacion, ya
464 #se corrige en esta parte)
465 posJ = 'p'
466 posK = 'k'
467 posL = 'r'
468 posM = 'n'
469 posN = 'w'
470 posO = 't'
471 posP = 'bhqvzx' # 0
472 posQ = 'l'
473 posR = 'u'
474 posS = 'bhqvzx' # 0
475 posT = 'e'
476 posU = 'bhqvzx'
477 posV = 'y'
478 posW = 'j'
479 posX = 'a'
480 posY = 'm'
481 posZ = 'c'
482 c = 0
483
484 # funcion para definir nuestro nuevo alfabeto quitando los caracteres encontrados
485 nuevoAlfabeto = 'abcdefghijklmnopqrstuvwxyz'
486 caracteresEncontrados = ['o','s','f','d','i','g','p','k','r','n','w','t','l',
487                           'u','e','y','j','a','m','c']
488 for a in caracteresEncontrados:
489     nuevoAlfabeto = nuevoAlfabeto.replace(a, "")
490 print (nuevoAlfabeto) # bhqvzx
491
492 # probando en htigme
493 for p in dic:
494     if len(p) == 6:
495         if (p[0] in posH and p[1] in posT and p[2] in posI and p[3] in posG
496             and p[4] in posM and p[5] in posE):
497             c = c + 1
498             print(p)
499 print(c)
500 # 1 resultado: behind

```

Salida:

```
Drive already mounted at /content/drive;  
55900 palabras leidas  
bhkqvzx  
darkest  
1
```

Probando con: **htigmc**

[illegible]**Salida:**

```
Drive already mounted at /content/drive;  
55900 palabras leidas  
bhqvzx  
behind  
1
```

Probando con: **ybfgrmu**

[illegible]

Salida:

```
Drive already mounted at /content/drive;
55900 palabras leidas
gqvxx
moving
1
```

Ya tendríamos todo el abecedario

```
550 posA = 'X' # 0
551 posB = 'o'
552 posC = 's'
553 posD = 'f'
554 posE = 'd'
555 posF = 'v'
556 posG = 'i'
557 posH = 'b'
558 posI = 'h'
559 posJ = 'p'
560 posK = 'k'
561 posL = 'r'
562 posM = 'n'
563 posN = 'w'
564 posO = 't'
565 posP = 'X' # 0
566 posQ = 'l'
567 posR = 'u'
568 posS = 'X' # 0
569 posT = 'e'
570 posU = 'g'
571 posV = 'y'
572 posW = 'j'
573 posX = 'a'
574 posY = 'm'
575 posZ = 'c'
```

alfabeto = abcdefghijklmnopqrstuvwxyz

Llave = XosfdvibhpkrnwtXluXegyjamc

Finalmente usamos la función de descifraSustituye() mandando como parámetros la llave y la cadena (texto cifrado), y finalmente la función nos va a retornar el texto descifrado.

```
580 alfabetoFinal = "XosfdvibhpkrnwtXluXegyjamc"
581 print (descifraSustituye(cifrado,alfabetoFinal))
```

donde cifrado contiene el texto:

Oitlt xlt onb ovjtc bd jtbtqt nib ngqq otqq vbr oixo vbr zxmmbo yxkt x
egddtlmtzt gm oigc nblqe: oibct nib xlt xdlxge ob olv xme oibct nib xlt
xdlxge vbr ngqq crzztte Crzztcc gc qgkgmu vbrlctqd, qgkgmu nixo vbr eb,
xme qgkgmu ibn vbr eb go Nitmtftl vbr dgme vbrlctqd bm oit cget bd oit
yxwblgov, go gc ogyt ob jxrct xme ltdqtzo Kttj vbrl dxzt xqnxvc obnxle
oit crmcigmt xme cixebnc ngqq dxqq htigme vbr Gd vbr ixft tftlvoigmu
rmetl zbmolbq, vbr xlt mbo ybfgmu dxco tmbrui Go gc erlgmu brl exlktco
ybytmoc oixo nt yrco dbzrc ob ctt oit qguio

Que descifrado sería:

there are two types of people who will tell you that you cannot make a difference in this world: those who are afraid to try and those who are afraid you will succeed success is liking yourself, liking what you do, and liking how you do it whenever you find yourself on the side of the majority, it is time to pause and reflect keep your face always toward the sunshine and shadows will fall behind you if you have everything under control, you are not moving fast enough it is during our darkest moments that we must focus to see the light

Traducido al español:

hay dos tipos de personas que te dirán que no puedes marcar la diferencia en este mundo: los que tienen miedo de intentarlo y los que tienen miedo de que tengas éxito el éxito es gustarte a ti mismo, gustarte lo que haces y gustarte cómo lo haces siempre que te encuentres en el lado de la mayoría, es el momento de hacer una pausa y reflexionar mantén tu cara siempre hacia el sol y las sombras caerán detrás de ti si tienes todo bajo control, no te estás moviendo lo suficientemente rápido es durante nuestros momentos más oscuros cuando debemos centrarnos para ver la luz

TEXTO 2:

Texto cifrado:

Jz bzz qwjjd ped nevvd le vkzhd eo mod hdlvd kelwkhd odvmkdjw d neknd hwove naejzpwvke qe qelvdobd. Vmrvd jd bzod w qwjepvdvd odvmkdjpwovw qd gdkwve kznnezlw z kegeqe gwoqee, umeoqe gwkw jd pdffezk gdkvw qwfje doepdje ozo lzoz ownwlldekew fdccew, ow lgwnedje kwneolezoe. Lzjz rwjeoe dcevdov j'oend gdkvw naemld, naw ad qe rdjvz jd lvwllld vkelvwbd qwe jzkz znnae kdlwlfodve. Lzoz dggwod dkkehvdz eo Evdjed wq w mo jmzfz qdhhwkz eondovwhzjw. Ewke az helevdvz ej nwovkz qe Kzpd w qwhz qekw naw w qdhhwkz mod nevvd nzo pzjve pzompwove dovenae, eordvve lzoz keldjwove djw wzgnd Kzpdod. Az helvz ej Nzjzllwz w qwhz dppwvwwkw naw gwoldhz rzllw gem gennzjz, eohwnw w fkdoqw wq w qdhhwkz mo cwj pzompwovz, gwnndvz naw ozo lzoz kemlnevz dq wovkdkw.

(idioma detectado: Italiano)

Código:

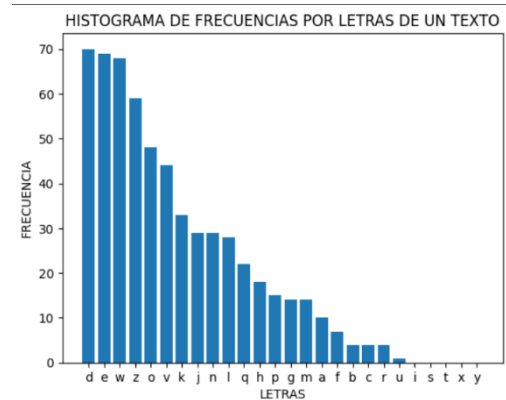
```
3 def cargaPalabras():
4     archivo = open('italiano_diccionario.txt', 'r')
5     renglon = archivo.readline()
6     palabras = renglon.split()
7     print(len(palabras), 'palabras leidas')
8     return palabras
9
10 def cargaCifrado():
11     archivo = open('cifrado.txt', 'r')
12     renglon = archivo.readline()
13     return renglon
14
15 def descifraSustituye(cadena, alfabetoLlave):
16     alfabeto = 'abcdefghijklmnopqrstuvwxyz'
17     nuevaCadena = ""
18     for letra in cadena:
19         if letra in alfabeto:
20             posicion = alfabetoLlave.find(letra)
21             nuevaCadena = nuevaCadena + alfabeto[posicion]
22         else:
23             nuevaCadena = nuevaCadena + letra
24     return nuevaCadena
25
26 dic = cargaPalabras()
27 cifrado = cargaCifrado()
28 histo = {}
29 alfabeto = 'abcdefghijklmnopqrstuvwxyz'
30 for e in alfabeto:
31     histo[e] = 0
32 for letra in cifrado:
33     if letra in alfabeto:
34         histo[letra] += 1
35 print(histo)
36
```

La salida del histograma del Texto cifrado 2, es el siguiente:

{'a': 10, 'b': 4, 'c': 4, 'd': 70, 'e': 69, 'f': 7, 'g': 14, 'h': 18, 'i': 0, 'j': 29, 'k': 33, 'l': 28, 'm': 14, 'n': 29, 'o': 48, 'p': 15, 'q': 22, 'r': 4, 's': 0, 't': 0, 'u': 1, 'v': 44, 'w': 68, 'x': 0, 'y': 0, 'z': 59}

En el texto cifrado 2, gracias al histograma de frecuencias, omitimos 5 posibles letras: i, s, t, x, y; además, como en el idioma italiano no existen las letras: J, K, W, X, Y, podemos omitirlas de las letras que buscamos, quedándonos de la siguiente manera:

```
posA="abcdefghijklmnopqrstuvwxyz"
posB="abcdefghijklmnopqrstuvwxyz"
posC="abcdefghijklmnopqrstuvwxyz"
posD="abcdefghijklmnopqrstuvwxyz"
posE="abcdefghijklmnopqrstuvwxyz"
posF="abcdefghijklmnopqrstuvwxyz"
posG="abcdefghijklmnopqrstuvwxyz"
posH="abcdefghijklmnopqrstuvwxyz"
posJ="abcdefghijklmnopqrstuvwxyz"
posK="abcdefghijklmnopqrstuvwxyz"
posL="abcdefghijklmnopqrstuvwxyz"
posM="abcdefghijklmnopqrstuvwxyz"
posN="abcdefghijklmnopqrstuvwxyz"
posO="abcdefghijklmnopqrstuvwxyz"
posP="abcdefghijklmnopqrstuvwxyz"
posQ="abcdefghijklmnopqrstuvwxyz"
posR="abcdefghijklmnopqrstuvwxyz"
posU="abcdefghijklmnopqrstuvwxyz"
posV="abcdefghijklmnopqrstuvwxyz"
posW="abcdefghijklmnopqrstuvwxyz"
posZ="abcdefghijklmnopqrstuvwxyz"
```



Probando con: **dppwvvkw**

```
38 #PROBANDO CON: dppwvvkw
39
40 posA="abcdefghijklmopqrstuvz"
41 posB="abcdefghijklmopqrstuvz"
42 posC="abcdefghijklmopqrstuvz"
43 posD="abcdefghijklmopqrstuvz"
44 posE="abcdefghijklmopqrstuvz"
45 posF="abcdefghijklmopqrstuvz"
46 posG="abcdefghijklmopqrstuvz"
47 posH="abcdefghijklmopqrstuvz"
48 posI="abcdefghijklmopqrstuvz"
49 posK="abcdefghijklmopqrstuvz"
50 posL="abcdefghijklmopqrstuvz"
51 posM="abcdefghijklmopqrstuvz"
52 posN="abcdefghijklmopqrstuvz"
53 posO="abcdefghijklmopqrstuvz"
54 posP="abcdefghijklmopqrstuvz"
55 posQ="abcdefghijklmopqrstuvz"
56 posR="abcdefghijklmopqrstuvz"
57 posU="abcdefghijklmopqrstuvz"
58 posV="abcdefghijklmopqrstuvz"
59 posW="abcdefghijklmopqrstuvz"
60 posZ="abcdefghijklmopqrstuvz"
61
62 c=0
63 for p in dic:
64     if len(p)==9:
65         if p[1]==p[2] and p[3]==p[6] and p[3]==p[-1] and p[4]==p[5]:
66             c+=1
67             print(p)
68 print("Num palabras: ",c)
69
70
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

affannata
allessere
ammettere
annettere
appannata
attaccata
avvizziti
eccellere
occorrono
Num palabras: 10
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano>

Palabras Validas:

- allessere
- ammettere
- annettere
- avvizziti

Reducción de Posibles combinaciones:

- posD: a
- posP: lmnv
- posW: ei
- posV: stz
- posK: rt

Probando con: **ownwlldkew**

```
71  #PROBANDO CON: ownwlldkew
72
73  posA="abcdefghijklmnopqrstuvz"
74  posB="abcdefghijklmnopqrstuvz"
75  posC="abcdefghijklmnopqrstuvz"
76  posD="a"
77  posE="abcdefghijklmnopqrstuvz"
78  posF="abcdefghijklmnopqrstuvz"
79  posG="abcdefghijklmnopqrstuvz"
80  posH="abcdefghijklmnopqrstuvz"
81  posJ="abcdefghijklmnopqrstuvz"
82  posK="rt"
83  posL="abcdefghijklmnopqrstuvz"
84  posM="abcdefghijklmnopqrstuvz"
85  posN="abcdefghijklmnopqrstuvz"
86  posO="abcdefghijklmnopqrstuvz"
87  posP="lmnv"
88  posQ="abcdefghijklmnopqrstuvz"
89  posR="abcdefghijklmnopqrstuvz"
90  posU="abcdefghijklmnopqrstuvz"
91  posV="stz"
92  posW="ei"
93  posX="abcdefghijklmnopqrstuvz"
94
95  c=0
96  for p in dic:
97      if len(p)==10:
98          if p[1] in posW and p[1]==p[3] and p[1]==p[-1] and p[-4]=='a' and p[4]==p[5]:
99              c+=1
100             print(p)
101  print("Num palabras: ",c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
necessarie
Num palabras: 1
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano>
```

Palabras Validas:

- necessarie

Reducción de Posibles combinaciones:

- posO: n
- posW: e
- posN: c
- posL: s
- posD: a
- posK: r
- posE: i

Probando con: **Nzjzllwz**

```
138 #PROBANDO CON: Nzjzllwz
139
140 posA="abcdefghilmnopqrstuvz"
141 posB="abcdefghilmnopqrstuvz"
142 posC="abcdefghilmnopqrstuvz"
143 posD="a"
144 posE="i"
145 posF="abcdefghilmnopqrstuvz"
146 posG="abcdefghilmnopqrstuvz"
147 posH="abcdefghilmnopqrstuvz"
148 posJ="abcdefghilmnopqrstuvz"
149 posK="r"
150 posL="s"
151 posM="abcdefghilmnopqrstuvz"
152 posN="c"
153 posO="n"
154 posP="lmnv"
155 posQ="abcdefghilmnopqrstuvz"
156 posR="abcdefghilmnopqrstuvz"
157 posU="abcdefghilmnopqrstuvz"
158 posV="stz"
159 posW="e"
160 posZ="abcdefghilmnopqrstuvz"
161
162 c=0
163 for p in dic:
164     if len(p)==8:
165         if p[0]=='c' and p[1]==p[3] and p[1]==p[-1] and p[-3]=='s' and p[4]==p[5]:
166             c+=1
167             print(p)
168 print("Num palabras: ",c)
169
170
171
```

colosseo

Num palabras: 1

PS C:\Users\JosueManzano\Desktop\DescifradoItaliano>

Palabras validas:

- colosseo

Reducción de Posibles combinaciones:

- posZ: o
- posJ: l

Probando con: **gennzjz**

```
138 #PROBANDO CON: gennzjz
139
140 posA="bdfghmpqtuvz"
141 posB="bdfghmpqtuvz"
142 posC="bdfghmpqtuvz"
143 posD="a"
144 posE="i"
145 posF="bdfghmpqtuvz"
146 posG="bdfghmpqtuvz"
147 posH="bdfghmpqtuvz"
148 posJ="l"
149 posK="r"
150 posL="s"
151 posM="bdfghmpqtuvz"
152 posN="c"
153 posO="n"
154 posP="mv"
155 posQ="bdfghmpqtuvz"
156 posR="bdfghmpqtuvz"
157 posU="bdfghmpqtuvz"
158 posV="tz"
159 posW="e"
160 posZ="o"
161
162 c=0
163 for p in dic:
164     if len(p)==7:
165         if p[-1]==p[-3] and p[-1]=='o' and p[2]==p[3] and p[2]=='c' and p[-2]=='l' and p[1]=='i':
166             c+=1
167             print(p)
168 print("Num palabras: ",c)
169
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
niccolo
piccolo
Num palabras: 2
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano> 
```

Palabras validas:

- niccolo
- piccolo

Reducción de Posibles combinaciones:

- posG: p

Probando con: **odvmkdjpwovw**

```

172 #PROBANDO CON: odvmkdjpwovw
173 posA="bdfghmqtuvz"
174 posB="bdfghmqtuvz"
175 posC="bdfghmqtuvz"
176 posD="a"
177 posE="i"
178 posF="bdfghmqtuvz"
179 posG="p"
180 posH="bdfghmqtuvz"
181 posI="l"
182 posK="r"
183 posL="s"
184 posM="bdfghmqtuvz"
185 posN="c"
186 posO="n"
187 posP="mv"
188 posQ="bdfghmqtuvz"
189 posR="bdfghmqtuvz"
190 posU="bdfghmqtuvz"
191 posV="tz"
192 posW="e"
193 posZ="o"
194 #PROBANDO CON: odvmkdjpwovw
195 c=0
196 for p in dic:
197     if len(p)==12:
198         if p[0]=='n' and p[4]=='r' and p[1]=='a' and p[-4]==p[-1] and p[-1]=='e':
199             c+=1
200             print(p)
201 print("Num palabras: ",c)
202

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

naturalmente

Num palabras: 1

PS C:\Users\JosueManzano\Desktop\DescifradoItaliano>

Palabras validas:

- naturalmente

Reducción de Posibles combinaciones:

- posV: t
- posM: u
- posP: m

Probando con: **qelvdbbd**

```
207 #PROBANDO CON: qelvdbbd
208 posA="bdfghmqt"
209 posB="bdfghmqt"
210 posC="bdfghmqt"
211 posD="a"
212 posE="i"
213 posF="bdfghmqt"
214 posG="p"
215 posH="bdfghmqt"
216 posJ="l"
217 posK="r"
218 posL="s"
219 posM="u"
220 posN="c"
221 posO="n"
222 posP="m"
223 posQ="bdfghmqt"
224 posR="bdfghmqt"
225 posU="bdfghmqt"
226 posV="t"
227 posW="e"
228 posZ="o"
229 #PROBANDO CON: qelvdbbd
230 c=0
231 for p in dic:
232     if len(p)==8:
233         if p[1]=='i' and p[3]=='t' and p[2]=='s' and p[-4]==p[-1] and p[-3]=='r' and p[-1]=='a':
234             c+=1
235             print(p)
236 print("Num palabras: ",c)
237
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
distanza
Num palabras: 1
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano> 
```

Palabras validas:

- distanza

Reducción de Posibles combinaciones:

- posQ: d
- posB: z

Probando con: **dovenae**

```
240 #PROBANDO CON: dovenae
241 posA="bfgqhv"
242 posB="z"
243 posC="bfgqhv"
244 posD="a"
245 posE="i"
246 posF="bfgqhv"
247 posG="p"
248 posH="bfgqhv"
249 posI="l"
250 posK="r"
251 posL="s"
252 posM="u"
253 posN="c"
254 posO="n"
255 posP="m"
256 posQ="d"
257 posR="bfgqhv"
258 posU="bfgqhv"
259 posV="t"
260 posW="e"
261 posZ="o"
262 #PROBANDO CON: dovenae
263 c=0
264 for p in dic:
265     if len(p)==7:
266         if p[0]=='a' and p[1]=='n' and p[2]=='t' and p[-4]==p[-1] and p[-1]=='i' and p[-2] in posA:
267             c+=1
268             print(p)
269 print("Num palabras: ",c)
270
271
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
antichi
Num palabras: 1
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano> |
```

Palabras validas:

- antichi

Reducción de Posibles combinaciones:

- posA: h

Probando con: **umeoqe**

```
274 #PROBANDO CON: umeoqe
275 posA="h"
276 posB="z"
277 posC="bfgqv"
278 posD="a"
279 posE="i"
280 posF="bfgqv"
281 posG="p"
282 posH="bfgqv"
283 posI="l"
284 posK="r"
285 posL="s"
286 posM="u"
287 posN="c"
288 posO="n"
289 posP="m"
290 posQ="d"
291 posR="bfgqv"
292 posU="bfgqv"
293 posV="t"
294 posW="e"
295 posZ="o"
296 #PROBANDO CON: umeoqe
297 c=0
298 for p in dic:
299     if len(p)==6:
300         if p[0] in posU and p[1]=='u' and p[2]=='i' and p[-4]==p[-1] and p[-1]=='i':
301             c+=1
302             print(p)
303 print("Num palabras: ",c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
quinci
quindi
quinti
Num palabras: 3
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano> |
```

Palabras validas:

- quinci
- quindi
- quinti

Reducción de Posibles combinaciones:

- posU: q

Probando con: **dkkehdivz**

```
308 #PROBANDO CON: dkkehdivz
309 posA="h"
310 posB="z"
311 posC="bfgv"
312 posD="a"
313 posE="i"
314 posF="bfgv"
315 posG="p"
316 posH="bfgv"
317 posJ="l"
318 posK="r"
319 posL="s"
320 posM="u"
321 posN="c"
322 posO="n"
323 posP="m"
324 posQ="d"
325 posR="bfgv"
326 posU="q"
327 posV="t"
328 posW="e"
329 posZ="o"
330 #PROBANDO CON: dkkehdivz
331 c=0
332 for p in dic:
333     if len(p)==8:
334         if p[0]=='a' and p[1]=='r' and p[3]=='i' and p[1]==p[2] and p[-1]=='o':
335             c+=1
336             print(p)
337 print("Num palabras: ",c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

arrivato

Num palabras: 1

PS C:\Users\JosueManzano\Desktop\DescifradoItaliano>

Palabras validas:

- arrivato

Reducción de Posibles combinaciones:

- posH: v

Probando con: **qwffe**

```
340
341 #PROBANDO CON: qwffe
342 posA="h"
343 posB="z"
344 posC="bfg"
345 posD="a"
346 posE="i"
347 posF="bfg"
348 posG="p"
349 posH="v"
350 posJ="l"
351 posK="r"
352 posL="s"
353 posM="u"
354 posN="c"
355 posO="n"
356 posP="m"
357 posQ="d"
358 posR="bfg"
359 posU="q"
360 posV="t"
361 posW="e"
362 posZ="o"
363 #PROBANDO CON: qwffe
364 c=0
365 for p in dic:
366     if len(p)==5:
367         if p[0]=='d' and p[1]=='e' and p[-2]=='l' and p[-1]=='i':
368             c+=1
369             print(p)
370 print("Num palabras: ",c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
degli
delli
Num palabras: 2
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano>
```

Palabras validas:

- degli
- delli

Reducción de Posibles combinaciones:

- posF: g

Probando con: **dcevdoz**

```
374 #PROBANDO CON: dcevdoz
375 posA="h"
376 posB="z"
377 posC="bf"
378 posD="a"
379 posE="i"
380 posF="g"
381 posG="p"
382 posH="v"
383 posJ="l"
384 posK="r"
385 posL="s"
386 posM="u"
387 posN="c"
388 posO="n"
389 posP="m"
390 posQ="d"
391 posR="bf"
392 posU="q"
393 posV="t"
394 posW="e"
395 posZ="o"
396 #PROBANDO CON: dcevdoz
397 c=0
398 for p in dic:
399     if len(p)==7:
400         if p[0]=='a' and p[2]=='i' and p[0]==p[-3] and p[-2]=='n' and p[-1]=='o':
401             c+=1
402             print(p)
403 print("Num palabras: ",c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
abitano
Num palabras: 1
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano> 
```

Palabras validas:

- abitano

Reducción de Posibles combinaciones:

- posC: b
- posR: f

Las posibles letras nos quedan de la siguiente forma:

```
posA="h"  
posB="z"  
posC="b"  
posD="a"  
posE="i"  
posF="g"  
posG="p"  
posH="v"  
posJ="l"  
posK="r"  
posL="s"  
posM="u"  
posN="c"  
posO="n"  
posP="m"  
posQ="d"  
posR="f"  
posU="q"  
posV="t"  
posW="e"  
posZ="o"
```

```
posA="h" posB="z" posC="b" posD="a" posE="i" posF="g" posG="p" posH="v"  
posJ="l"  
  
posK="r" posL="s" posM="u" posN="c" posO="n" posP="m" posQ="d" posR="f"  
posU="q"  
  
posV="t" posW="e" posZ="o"
```

Ahora vamos armando nuestra llave. Para armar la llave escribimos el alfabeto:

abcdefghijklmnopqrstuvwxyz

Y para cada letra buscamos su original, por ejemplo, en el caso de la "a":

```
posA="h" posB="z" posC="b" posD="a" posE="i" posF="g" posG="p" posH="v"  
posJ="l"  
  
posK="r" posL="s" posM="u" posN="c" posO="n" posP="m" posQ="d" posR="f"  
posU="q"  
  
posV="t" posW="e" posZ="o"
```

Vemos que en el texto cifrado apareció una “D” en sustitución de la “a”, por lo que la escribimos

debajo de la “a”:

abcdefghijklmnopqrstuvwxyz

d

Ahora buscamos a “b” y reemplazamos, y así sucesivamente, excepto con las letras que no aparecen, donde podemos poner una X:

posA="h" posB="z" posC="b" posD="a" posE="i" posF="g" posG="p" posH="v"
posJ="l"

posK="r" posL="s" posM="u" posN="c" posO="n" posP="m" posQ="d" posR="f"
posU="q"

posV="t" posW="e" posZ="o"

Alfabeto:

abcdefghijklmnopqrstuvwxyz

Llave:

dcnqwrfaeXXjpozguklvmhXXXb

Finalmente usamos la función de **descifraSustituye()** mandando como parámetros la llave y la cadena (texto cifrado), y finalmente la función nos va a retornar el texto descifrado.

```
39
40 llave='dcnqwrfaeXXjpozguklvmhXXXb'
41 cadena="Jz bzz qwjdd ped nevvd le vkzhd eo mod hdlvd kelwkh odvmkdjw d neknd hwove naejzpwvke qe qelvdbod. \"
42 \"Vmvvd jd bzod w qwjepdvdd odvmkdjpwovw qd gdwve kznnezlw z kegeqe gwoqee, umeoqe gwkw jd pdfpezk \"
43 \"gdkvw qwfje doepdje ozo lzo owmllldkew fdccew, ow lgwnedje kwneolezoe. Lzjz rwjeoe dcevdz j'oend \"
44 \"gdkvw naemld, naw ad qe rdjvz jd lvwllld vkelvwbdd qwe jzkz znnae kdlwfodve. Lzo dggwod dkkehdyz eo \"
45 \"Evdjed wq w mo jmfz qdhhwkz eondovhzhjw. Ewke az helevdyz ej nwovkz qe Kzpd w qwhz qekw naw w qdhhwkz \"
46 \"mod nevvd nzo pzjve pzompwove dovenae, eordvve lzo keldjwove djw wgznd Kzpdod. Az helvz ej Nzjzllwz w qwhz \"
47 \"dppwvwkw naw gwoldhz rzllw gem gennjz, eohnmw w fkdoqw wq w qdhhwkz mo cwj pzompwovz, gwndvz naw ozo lzo kemlnevz dq wovdkw.\"
48 b=descifraSustituye(cadena, llave)
49 print(\"\\nTexto Descifrado por Sustitucion: \")
50 print(b)
```

Salida:

```
Texto Descifrado por Sustitucion:
Jo zoo della mia citta si trova in una vasta riserva naturale a circa venti chilometri di distanza. Vutta la zona e delimitata naturalmente da pareti ro
cciose o ripidi pendii, quindi per la maggior parte degli animali non sono necessarie gabbie, ne speciali recinzioni. Lolo felini abitano l'nica parte c
hiosa, che ha di fatto la stessa tristezza dei loro occhi rassegnati. Lono appena arrivato in Italia ed e un luogo davvero incantevole. Eeri ho visitato
il centro di Roma e devo dire che e davvero una citta con molti monumenti antichi, infatti sono risalenti all epoca Romana. Ao visto il Nolosseio e devo
ammettere che pensavo fosse piu piccolo, invece e grande ed e davvero un bel monumento, peccato che non sono riuscito ad entrare.
PS C:\Users\JosueManzano\Desktop\DescifradoItaliano>
```


Texto Descifrado del italiano:

lo zoo della mia città si trova in una vasta riserva naturale a circa venti chilometri di distanza. tutta la zona è delimitata naturalmente da pareti rocciose o ripidi pendii, quindi per la maggior parte degli animali non sono necessarie gabbie, né speciali recinzioni. solo felini abitano l'unica parte chiusa, che ha di fatto la stessa tristezza dei loro occhi rassegnati. sono appena arrivato in Italia ed è un luogo davvero incantevole. ieri ho visitato il centro di Roma e devo dire che è davvero una città con molti monumenti antichi, infatti sono risalenti all'epoca romana. ho visto il Colosseo e devo ammettere che pensavo fosse più piccolo, invece è grande ed è davvero un bel monumento, peccato che non sono riuscito ad entrare.

Texto descifrado traducido al español:

El zoológico de mi ciudad está situado en una gran reserva natural a unos veinte kilómetros de distancia. Toda la zona está delimitada naturalmente por paredes rocosas o pendientes pronunciadas, por lo que para la mayoría de los animales no son necesarias jaulas ni recintos especiales. sólo felinos habitan la única parte cerrada, que tiene la misma tristeza que sus ojos resignados. Acabo de llegar a Italia y es un lugar verdaderamente encantador. ayer visité el centro de Roma y debo decir que verdaderamente es una ciudad con muchos monumentos antiguos, de hecho datan de la época romana. Vi el Coliseo y debo admitir que pensé que era más pequeño, pero en cambio es grande y es realmente un monumento hermoso, lástima que no pude entrar.