

Message decryption

Message encryption is necessary to ensure safe transmission of data.

Messages can be encrypted by changing the order of the characters in the message.

Consider the example: 'Luke, I am your father'

The encryption is done following these steps:

- An integer number **c** is chosen at random. (say $c=3$)
- The message is then written in a **$n \times c$ array** (extra characters '&' might be needed if the string is too short). The dimension **n** is determined by the length of the string and the value of **c**. So the corresponding array for our message looks like this:

L	u	k
e	,	
I		a
m		y
o	u	r
	f	a
t	h	e
r	&	&

- The encrypted message is obtained from the transpose array (after removing the extra characters '&'):

L	e	I	m	o		t	r
u	,			u	f	h	&
k		a	y	r	a	e	&

Encrypted message: 'Lelmo tru, ufhk ayrae'

Project

Each file listed below contains text (a fragment of a book in English) encrypted using the method described above choosing c at random in the $[25,100]$ interval.

Write a function **decrypt('txtfile')** that decrypts the file corresponding to your UB Person Number modulo 30 and print the decrypted message. The function should take as a variable the name of your .txt file.

A dictionary of English words is available here [!\[\]\(0f848bbd71cef6b345273b16f905912a_img.jpg\) dictionary](#)

Note. This is a programming project. Your project report does not need include narrative, beyond comments explaining how your code works. The project will be graded according to the following rubrics:

- Code that successfully decrypts the text file: 70%
- Report organization and code documentation: 30%

0. [!\[\]\(c6a8736a601a632e2c96605cf66055ed_img.jpg\) text0.txt](#)
1. [!\[\]\(64ef2b19d70b31fbbfce0e0e2aa3d7b4_img.jpg\) text1.txt](#)
2. [!\[\]\(9ba1c633ca37327550476fd7d0d00348_img.jpg\) text2.txt](#)
3. [!\[\]\(9123a11efb62a56709757215846100c3_img.jpg\) text3.txt](#)
4. [!\[\]\(81ed9b526bb4d794d4b41c492b59462a_img.jpg\) text4.txt](#)
5. [!\[\]\(73ff443d232f42a7f7c62fc0b625b197_img.jpg\) text5.txt](#)
6. [!\[\]\(1875783f027eea357c44cf6b28874dc9_img.jpg\) text6.txt](#)
7. [!\[\]\(8e3c070538b017ee3f9ae5ec34168d7e_img.jpg\) text7.txt](#)
8. [!\[\]\(65ccf85a896fb6b80747b2759e07d8eb_img.jpg\) text8.txt](#)
9. [!\[\]\(a5237a020a008d2d5cf7735d226bf122_img.jpg\) text9.txt](#)
10. [!\[\]\(7eb94c6063e3d1d20ffe05d53e8af840_img.jpg\) text10.txt](#)
11. [!\[\]\(44ba307df15eee601446fc825a5db580_img.jpg\) text11.txt](#)
12. [!\[\]\(598afa8e1838a908a1e512a1385fcf4f_img.jpg\) text12.txt](#)
13. [!\[\]\(d52f73ccd8d44f75ecf29f6f349bdf72_img.jpg\) text13.txt](#)
14. [!\[\]\(2abc45a71a59ab2ca88e069f7fcf18b0_img.jpg\) text14.txt](#)
15. [!\[\]\(6e072688b6196c6b1296fe6d1f9b4091_img.jpg\) text15.txt](#)
16. [!\[\]\(8ea9e748694fec1787938840a6d6b266_img.jpg\) text16.txt](#)
17. [!\[\]\(443359d7483d497de9fd555f7d7299fc_img.jpg\) text17.txt](#)
18. [!\[\]\(482824bd29233798109f56f904c16314_img.jpg\) text18.txt](#)
19. [!\[\]\(f8a9a761357a87ee098d4a10e3e157d4_img.jpg\) text19.txt](#)
20. [!\[\]\(f818647475e1c3785e5d4cfa4137a8f7_img.jpg\) text20.txt](#)
21. [!\[\]\(79a38e084fe1569bc70af48346e10cda_img.jpg\) text21.txt](#)

22.  text22.txt

23.  text23.txt

24.  text24.txt

25.  text25.txt

26.  text26.txt

27.  text27.txt

28.  text28.txt

29.  text29.txt