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 x 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: 'LeImo 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 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. text0.txt
1. text1.txt
2. text2.txt
3. text3.txt
4. text4.txt
5. text5.txt
6. text6.txt
7. text7.txt
8. text8.txt
9. text9.txt
10. text10.txt
11. text11.txt
12. text12.txt
13. text13.txt
14. text14.txt
15. text15.txt
16. text16.txt
17. text17.txt
18. text18.txt
19. text19.txt
20. text20.txt
21. 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