

2019 年 06 月 07 日

## 文字列 – 回文、シーザー暗号

### 課題 9：回文

ファイルを読み込み、語句を一つずつ検証した。その前、`strip()`でスペースや改行を外した。

回文である語句を出力した。下記の表 1 に示す。合計で 57 個である。

表 1： 課題 9、出力した回文の語句

aga	ere	noon	rotator
aha	eve	nun	rotor
ala	ewe	pap	sagas
ana	eye	peep	sees
anna	gag	pep	sexes
bib	gig	pip	shahs
bob	huh	poop	sis
bub	kayak	pop	sos
civic	level	pup	tat
dad	madam	radar	tenet
deed	minim	redder	tit
deified	mom	refer	toot
did	mum	repaper	tot
dud	non	reviver	wow
eke			

また、作った回文検証の関数は特別な場合には、何を出力するか、下記の表 2 に示す。一つの文字の語句は常に回文である。アルファベットではない文字の場合、例えば数字も回文として検証している。複数の同じ文字が連続する場合にも正しく検証する（ループ内で `strip(○)` を使ったら、そうではないのである）。

表 2： 課題 9、回文検証の関数、特別な場合の出力

入力	出力
a	True
12321	True
aaaaava	False

## 課題 10：シーザー暗号

先生のスライドの「メッセージの符号化」のコードに基づいて、シーザー暗号化のプログラムを作成した。暗号化の場合、鍵はランダムに生成する。復号化の場合、鍵は入力される。入力されたデータは、チェックされた後、暗号化・復号化されて、出力される。

表 3： 課題 10、様々なテストケース

プログラムの流れ	コメント
Press 0 for encode, 1 for decode: 0 Please input a message: Hello! My name is Tom. Encoded message using cipher [76]: CZggj75Ht5iVhZ5dn5Ojh'	普通の暗号化の例
Press 0 for encode, 1 for decode: 1 Please input a message: CZggj75Ht5iVhZ5dn5Ojh' Please input the cipher: 76 Decoded message using cipher [76]: Hello! My name is Tom.	普通の復号化の例 上記の逆転
Press 0 for encode, 1 for decode: 0 Please input a message: Is this working? #test #テスト Your message contains illegal letters: テ, ス, ト, #	違法な入力
Press 0 for encode, 1 for decode: 1 Please input a message: ABCabc123! Please input the cipher: -2 Decoded message using cipher [-2]: CDEcde345&	復号化の時、負の数を入力すると、好きな鍵を選べる
Press 0 for encode, 1 for decode: 1 Please input a message: CDEcde345& Please input the cipher: 2 Decoded message using cipher [2]: ABCabc123!	上記の逆転

課題 10-2 には、全てのカギを用いて（1～80, N=81）、もらったファイルの内容を復号化してみた。結果を他のファイルに書いた。それを見ると、正しい鍵は 40 であるということがわかった。下記の表 4 にはその結果を示す。

表 4： 課題 10－2 の結果

---

Decoded text using cipher [40]:

---

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum's long influence on Python is reflected in the title given to him by the Python community: Benevolent Dictator For Life (BDFL) - a post from which he gave himself permanent vacation on July 12, 2018.

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go transcompiler to improve performance under concurrent workloads.