Zadanie Szyfr Beauforta –LOGIA (2020/21), etap 3

Treść zadania

Joasia interesuje się szyframi, ostatnio czytała o szyfrze Beauforta. Szyfruje małe litery alfabetu łacińskiego biorąc najpierw odbicie symetryczne danej litery, tzn. zamiast litery **a** bierze literę **z**, zamiast litery **b** literę **y** itd. Następnie przesuwa cyklicznie (tzn. po literze **z** występuje ponownie litera **a**) symetryczne odbicie o określoną kluczem liczbę pozycji. Kluczem jest napis złożony z małych liter alfabetu łacińskiego, przy czym litera **a** – oznacza przesunięcie o 1, **b** – przesunięcie o 2, …, a **z** przesunięcie o 26. Gdy wykorzysta wszystkie litery klucza, zaczyna analizować klucz od początku.

wiadomość	k	o	n	k	u	r	s
odbicie symetryczne	p	l	m	p	f	i	h
klucz	l	o	g	i	a	l	o
przesunięcie	12	15	7	9	1	12	15
szyfrogram	b	a	t	y	g	u	w

Na przykład wiadomość *konkurs* zaszyfrowana kluczem *logia* to szyfrogram *batyguw*. Pomóż Joasi i napisz program, który wczyta wiadomość i klucz, a następnie zaszyfruje wiadomość podanym kluczem.

Wejście:

Dwa niepuste napisy oddzielone spacją o maksymalnej długości 500 znaków każdy. Pierwszy to wiadomość do szyfrowania, a drugi to klucz. Oba napisy zawierają wyłącznie małe litery alfabetu łacińskiego.

Wyjście:

Jedyny wiersz wyjścia to napis - zaszyfrowana wiadomość z wejścia podanym kluczem i metodą opisaną w treści zadania.

	Przykład 1	*Przykład 2*	*Przykład 3*
Wejście	**konkurs logia**	**spotkanie wtorek**	**scisle tajne**
Wyjście	**batyguw**	**eeayukjlk**	**bybvtp**

Omówienie rozwiązania

Szyfr nosi nazwę swego twórcy, czyli admirała Francisa Beauforta. Opracował on także używaną do dziś skalę siły wiatru. Szyfr jest przykładem szyfru samoodwrotnego, czyli zastosowanie procedury szyfrującej do zaszyfrowania szyfrogramu z tym samym kluczem da wiadomość. Szyfrowanie Cezara z przesunięciem alfabetu 26 literowego o 13 pozycji jest także samoodwrotne. Inny przykład szyfrowania samoodwrotnego znajdziemy w maszynach Enigma.

Implementując funkcję szyfrowania musimy umieć odwzorować literę na jej numer oraz odwrotnie – znając numer litery w alfabecie określić co to za litera. Możemy to zrobić definiując listę lub napis złożony z kolejnych liter alfabetu (przydatne m.in. gdy w alfabecie występują polskie znaki diakrytyczne

„ąćęłóśńźż") lub posługując się funkcjami **ord()** i **chr()**. Pierwsza zwraca kod ASCII danej litery, a druga literę o podanym kodzie ASCII. Litera **a** ma kod równy 97 a kolejne 25 małych liter alfabetu ma kody od 98 do 122. Aby otrzymać numery w zakresie od 0 do 25 należy odjąć od kodu ASCII litery wartość **ord('a')**.

Do szyfrowania kolejnych liter wiadomości używa się kolejnych liter z klucza. W tym celu wartość zmiennej `i` będzie powiększana o 1 po zaszyfrowaniu litery, a wartość wyrażenia **i mod długość_klucza** posłuży do odczytywania odpowiedniej wartości z klucza.

W pseudokodzie krok szyfrowania pojedynczej litery mógłby wyglądać tak:

```
1.    kod_litery ← (ord(klucz[i mod len(klucz)]) - ord(litera)) mod 26
2.    szyfrogram ← szyfrogram + chr(ord ('a') + kod_litery)
3.    i ← i + 1
```

Milcząco zakładamy, że operacja modulo zwraca dodatni wynik z zakresu 0..25 dla ujemnych argumentów. W języku Python rzeczywiście tak jest, lecz np. w C++ inaczej jest określany znak wyniku. W linii 1 nie wykonujemy przesunięcia kodu o wartość **ord('a')**, gdyż mamy tam odejmowanie, co powoduje, że te przesunięcia skracają się. Cały kod szyfrujący tekst w zmiennej **wiadomosc** przy pomocy klucza **klucz**:

```
1.    i ← 0
2.    dl = len(klucz)
3.    szyfrogram ← ''
4.    dopóki i < len(wiadomosc):
5.        litera ← wiadomosc[i]
6.        kod_litery ← (ord(klucz[i mod dl]) - ord(litera)) mod 26
7.        szyfrogram ← szyfrogram + chr( ord ('a') + kod_litery )
8.        i ← i + 1
9.    wynik szyfrogram
```

Rozwiązanie w języku Python

Kod źródłowy rozwiązania w języku Python może być następujący:

```
1.  def beaufort(wiadomosc, klucz):
2.      szyfrogram = ['']*len(wiadomosc)
3.      for i, litera in enumerate(wiadomosc):
4.          kod_litery = (ord(klucz[i % len(klucz)])-ord(litera)) % 26
5.          szyfrogram[i] = chr(ord ('a') + kod_litery)
6.      return ''.join(szyfrogram)
7.
8.  szyfruj = input().split()
9.  print(beaufort(szyfruj[0], szyfruj[1]))
```

Z uwagi na to, że napisy w języku Python są niezmienne (ang. immutable) i po każdym dodaniu litery do szyfrogramu przepisywany byłby w pamięci cały napis, to szyfrogram będzie tworzony w liście. W linii 2 tworzymy listę złożoną z pustych napisów. Lista ta ma długość równą liczbie znaków w wiadomości. Do numerowania poszczególnych liter w wiadomości użyjemy funkcji **enumerate()**. W zmiennej `i` w kolejnych iteracjach będzie zapisywany numer litery, a w zmiennej **litera** litera o indeksie `i` parametru **wiadomosc**. W linii 4 i 5 jest obliczany kod zaszyfrowanej litery zgodnie z opisem w pseudokodzie. Na końcu funkcji litery będące elementami listy zostaną złączone w napis przy pomocy metody **join()**. Metoda ta jest wywołana na rzecz napisu pustego z argumentem będącym listą. Użycie metody **split()** w linii 8 spowoduje, że napis wczytany przez funkcję

`input()` zostanie przekonwertowany na listę. Domyślnie znakiem dzielącym napis na poszczególne elementy listy jest spacja.

Testy

Testy, na których testowano rozwiązania rozróżniały różne długości wiadomości i klucza.

Grupa testów	Wejście	Wynik
I	aaa a	aaa
	xyz bb	edc
II	cezar c	aydcl
	cezar kleopatra	ihfoy
III	stereo stereo	aaaaaa
	kryptografia matematyka	cjvptmnhkvea
IV	czytomozliwe xyz	vzbeknjzopcv
	abcdefghijklmnopqrstuvwxyz bd	bczaxyvwturspqnolmjkhifgde
V	litwoojczyznomojatyjestesjakzdrowieileci etrzebacenictentylkosiedowiektociestracil pantadeusz	esuxmpvstbqnzhmuebuqliupiuektwymrl wvtqqrlhwuwcesomhyupnkgjilxsjqmhw qigbyfpiknuqre
	loremipsumdolorsitametconsecteturadipi scingelitseddoeiusmodtemporincididuntu tlaboreetdoloremagnaaliquautenimadmi nimveniamquisnostrudexercitationullamc olaborisnisiutaliquipexeacommodoconse quatduisauteiruredolorinreprehenderitin voluptatevelitessecillumdoloreeufugiatnu llapariaturexcepteursintoccaecatcupidata tnonproidentsuntinculpaquiofficiadeseru ntmollitanimidestlaborumveroeosetaccus amusetiustoodiodignissimosducimusquibl anditiispraesentiumvoluptatumdelenitiat quecorruptiquosdoloresetquasmolestiase xc epturisintoccaecaticupiditatenonprovide ntsimiliquesuntinculpaquiofficiadeserunt mollitiaanimiidestlaborumetdolorumfuga etharumquidemrerumfacilisestetexpedita distinctionamliberotemporecumsolutano bisesteligendioptiocumquenihilimpeditqu ominusidquodmaximeplaceatfacerepossi musomnisvoluptasassumendaestomnisdo lorrepellendustemporibusautemquibusda metautofficiisdebitisautrerumnecessitati bussaepeevenietutetvoluptatesrepudiand aesintetmolestiaenonrecusandaeitaquee arumrerumhicteneturacyceronogranicach dobraizla	tbcqfadqthlormnksaiqqwgpvbwrljvtyrln tvmllfcixdpyqbpgjlovqgiwwefuxxsgufsak flxbctlauheafmpzriwixrstasluratagvcllfm zxilwrlhobaeyhullvxbnxioyuplpkqfztieln qsismmaagavuzpocswramauktcydcdbos yfwhezaulhaaeyaadrmnjaudbcablqxzqxj veeuathriuiwfvmpjzoxvxqwliypxbkdimr qjpbgxklwpvfyxbudovfizbxhxnplzwybcer upycspmmugyogioyrlcqywnzvrogluvbya osypftosigdwmqvajabwdzbmturxazdrtw lfoibirdddsbjyneoubwaizsyaasvmacwtql fgqwfsgfhhbilsezbvsvwgvngwrxngbltgd mdciwmjbujkqzeptohepzonqtanjberlbai fulckyllexmkvotdxjwcyktikoowzwijaqvoy