

4. Szyfr Cezara (*)

Kurs programowania i algorytmiki OI: kurs.oi.edu.pl

Kod zadania: **szy**
Limit czasu: **1 s**
Limit pamięci: **256 MB**



W tym zadaniu Twój program będzie szyfrował i odszyfrowywał dane, używając algorytmu, który był znany już w czasach Juliusza Cezara. Szyfrowanie tekstu polega na szyfrowaniu kolejnych jego liter (pozostałe znaki pozostawiamy bez zmian). Każda litera zostaje zamieniona w k -tą następną w alfabecie (k jest stałą szyfru), przy czym jeżeli taka nie istnieje (wychodzimy za z), to odliczanie jest kontynuowane z powrotem od a . Szyfrowanie zachowuje wielkość liter (tj. małe przechodzą na małe, a wielkie na wielkie). Zakładamy, że w tekście występują jedynie litery alfabetu angielskiego (małe i wielkie) oraz znaki interpunkcyjne (bez spacji).

Dla przykładu, jeżeli $k = 5$, to małe litery tekstu przechodzą na małe litery szyfru według poniższej tabelki:

litera	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
szyfr	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e

Wielkie litery tekstu przechodzą na wielkie litery szyfru zgodnie z tą samą regułą.

Napisz program, który wczyta tekst do zaszyfrowania lub odszyfrowania i stałą k i zaszyfruje lub odszyfruje ten tekst, w zależności od polecenia.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę 1 lub 2. Liczba 1 oznacza „szyfruj”, a liczba 2 – „odszyfruj”. Drugi wiersz wejścia zawiera stałą k ($1 \leq k \leq 25$). Trzeci i ostatni wiersz wejścia zawiera tekst, złożony wyłącznie z liter (małych bądź dużych) i/lub znaków interpunkcyjnych (bez spacji). Tekst będzie zawierał co najmniej jeden znak i co najwyżej 10 000 znaków.

Wyjście

Pierwszy i jedyny wiersz wyjścia powinien zawierać tekst po zaszyfrowaniu bądź odszyfrowaniu.

Wejście dla testu szy0:

```
1
5
Szyfr_Cezara_jest_skuteczny!
```

Wyjście dla testu szy0:

```
Xedkw_Hjefwf_ojxy_xpzyjhed!
```

a dla danych wejściowych:

```
2
5
Xedkw_Hjefwf_ojxy_xpzyjhed!
```

poprawnym wynikiem jest:

```
Szyfr_Cezara_jest_skuteczny!
```

natomiast dla danych wejściowych:

```
1
24
xyz
```

poprawnym wynikiem jest:

```
vwx
```