

## Problem 10: Szyfr Cezara

**Punkty:** 20

**Autor:** Steve Gerali, Denver, Kolorado, Stany Zjednoczone

### Wprowadzenie do problemu

Szyfr Cezara to jeden z najwcześniejszych znanych nam szyfrów, a w dodatku jeden z najprostszych do nauki. To „szyfr podstawieniowy”, w którym każdą literę tekstu niezaszyfrowanego przesuwamy się o określoną liczbę miejsc do przodu w alfabecie. Na przykład, w przypadku przesunięcia równego 1, zamiast A podstawia się B, zamiast B C itd. Metoda została nazwana na cześć Juliusza Cezara, który stosował ją do komunikowania się ze swoimi generałami.

Aby możliwe było przekazanie zaszyfrowanej wiadomości, zarówno przekazujący, jak i odbierający muszą znać klucz szyfru, aby przekazujący mógł go zaszyfrować, a odbierający odszyfrować. W przypadku szyfru Cezara kluczem jest liczba liter, o którą należy przesunąć alfabet.

### Opis problemu

Pracujecie dla telewizji historycznej, która chce odszyfrować wszystkie komunikaty przesłane przez Juliusza Cezara do jego generałów, aby wykorzystać je w nowym dokumencie o rzymskim imperatorze. Otrzymacie listę zaszyfrowanych wiadomości i klucz, który prawdopodobnie został użyty do ich szyfrowania. Wasz program musi odszyfrować wiadomości.

Na potrzeby problemu będziemy stosować alfabet angielski, przedstawiony poniżej w standardowym porządku (z przesunięciem 0).

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

Po zaszyfrowaniu wiadomości z przesunięciem 1 każda litera w tekście niezaszyfrowanym zostanie zastąpiona odpowiednią literą podaną w poniższym alfabecie przesuniętym o 1.

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 A

Podczas odszyfrowywania wiadomości proces jest odwracany: litera w tekście zaszyfrowanym jest zastępowana odpowiednią literą z normalnego alfabetu angielskiego.

Spacje nie są szyfrowane i podczas odszyfrowywania pozostają na swoim miejscu.



## Przykładowe dane wejściowe

Pierwszy wiersz danych wejściowych waszego programu, **otrzymanego przez standardowe wejście**, będzie zawierać dodatnią liczbę całkowitą oznaczającą liczbę przypadków testowych. Każdy przypadek testowy będzie zawierać dwa wiersze:

- Wiersz z pojedynczą liczbą całkowitą oznaczającą klucz wiadomości - liczbę liter, o jaką alfabet został przesunięty podczas szyfrowania
- Wiersz zawierający małe litery i spacje, stanowiący zaszyfrowaną wiadomość

```
3
1
buubdl bu ebxo
3
ghvwurb wkh fdvwoh
6
yzkgr znk ynov
```

## Przykładowe dane wyjściowe

W każdym przypadku testowym wasz program powinien wyświetlić odszyfrowaną wiadomość. Wiadomości powinny być wyświetlane małymi literami, z zachowanymi spacjami.

```
attack at dawn
destroy the castle
steal the ship
```