

Zadanie: ZAK Zakłócenia [C]



POTYCZKI ALGORYTMICZNE

Potyczki Algoritmiczne 2021, runda druga. Limity: 512 MB, 1 s.

07.12.2021

W komunikacji sieciowej bardzo istotne jest, by korzystać z protokołów sieciowych, które zapewnią, że dane wysłane przez nadawcę wiadomości będą zgodne z danymi uzyskanymi przez odbiorcę wiadomości. Niedopilnowanie tego może skutkować trudnościami w poprawnym odczytaniu wiadomości przez odbiorcę. Taka sytuacja spotkała Bajtka i Bitka...

Bajtek chciał wysłać przez internet do Bitka ciąg n małych liter alfabetu angielskiego. W tym celu komputer Bajtka skonwertował każdą kolejną literę tego ciągu na jej ósmiobitową reprezentację w kodowaniu ASCII.*

Litera	Kod ASCII	Litera	Kod ASCII
a	01100001	n	01101110
b	01100010	o	01101111
c	01100011	p	01110000
d	01100100	q	01110001
e	01100101	r	01110010
f	01100110	s	01110011
g	01100111	t	01110100
h	01101000	u	01110101
i	01101001	v	01110110
j	01101010	w	01110111
k	01101011	x	01111000
l	01101100	y	01111001
m	01101101	z	01111010

Nietrudno zauważyć, że i -ta mała litera alfabetu angielskiego ($1 \leq i \leq 26$) otrzymuje kod ASCII, który jest zapisem binarnym liczby $96 + i$.

Następnie komputer Bajtka skleił n ósmiobitowych ciągów w jeden długi ciąg bitów długości $8n$ – reprezentację n -znakowego ciągu Bajtka. W końcu komputer Bajtka wysłał tę reprezentację przez internet do komputera Bitka. Niestety, każdy bit został wysłany w oddzielnym pakiecie sieciowym, przez co bity mogły dojść do Bitka w zupełnie innej kolejności niż oryginalna!

Przemieszany ciąg bitów dotarł wreszcie do Bitka. Oczywiście taki ciąg bitów nie musi obecnie reprezentować żadnego n -znakowego ciągu małych liter alfabetu angielskiego. Pomimo tego, że ciąg bitów nie zawiera żadnej dodatkowej informacji na temat ich poprawnej kolejności, Bitek postanowił się jednak nie poddawać. Uznał więc, że spróbuje odzyskać jakikolwiek ciąg n małych liter alfabetu angielskiego, którego reprezentacja w postaci ciągu $8n$ bitów mogła dojść do Bitka w takiej postaci. Pomóż mu i znajdź taki przykładowy ciąg – lub stwierdź, że taki nie istnieje!

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leq n \leq 10^5$).

W drugim wierszu wejścia znajduje się słowo binarne długości $8n$, które oznacza ciąg bitów otrzymany przez Bitka.

Wyjście

Jeśli ciąg otrzymany przez Bitka nie odpowiada żadnemu słowu długości n złożonemu wyłącznie z małych liter alfabetu angielskiego, to na wyjściu powinno znaleźć się pojedyncze słowo NIE.

W przeciwnym wypadku na wyjściu powinien znaleźć się dowolny ciąg znaków długości n złożony z małych liter alfabetu angielskiego, który po zakodowaniu przez komputer Bajtka i przesłaniu przez sieć do Bitka mógł dotrzeć jako ciąg bitów podany na wejściu. Jeśli istnieje wiele poprawnych ciągów znaków, możesz wypisać dowolny z nich.

*Więcej na temat kodowania ASCII możesz dowiedzieć się z Wikipedii: <https://pl.wikipedia.org/wiki/ASCII>. Jednakże podany tu opis w zupełności wystarczy do rozwiązania zadania.

Przykład

Dla danych wejściowych:

2
1100000011110111

poprawnym wynikiem jest na przykład:

ao

Natomiast dla danych wejściowych:

8
1011111010101100011011011010001010100011111111110001001001011010

poprawnym wynikiem jest na przykład:

potyczki

Zaś dla danych wejściowych:

1
00011000

poprawnym wynikiem jest:

NIE

Wyjaśnienie przykładu: Litera a w ASCII reprezentowana jest przez ciąg bitów 01100001, zaś litera o reprezentowana jest przez 01101111. Ciąg ao wysłany do Bitka został zatem przekonwertowany na ciąg 0110000101101111. Bity wiadomości mogły więc przyjść do komputera Bitka w kolejności 1100000011110111.

Podzadania

W niektórych grupach odpowiedź zawsze istnieje. Ponadto w tych grupach bity w trakcie transportu od Bajtka do Bitka zmieniały swoją kolejność jedynie w obrębie kodów ASCII do których należały. Innymi słowy, żaden bit nie wyszedł poza 8 pozycji, które reprezentowały jego oryginalną literę. Pomimo to, przy odtwarzaniu ciągu możesz dowolnie zamieniać bity miejscami.