

## Anagram: アナグラム

input ファイル “anagram.in”

output 標準出力

ソースファイル anagram.c/anagram.cpp/Anagram.java

時間制限 0.5 秒 / データ

ある文字列の文字の順序を並べ替えて別の文字列を作ること、また、そのように並べ替えて作られた文字列をアナグラム (Anagram) と呼ぶ。この問題では、一度も並べ替えをしていない文字列も元の文字列のアナグラムと呼ぶことにする。つまり、文字列 “EARTH” は文字列 “HEART” のアナグラムであり、文字列 “HEART” も、文字列 “HEART” のアナグラムである。

文字列が与えられると、多くの場合、その文字列のアナグラムは何通りも考えられる。たとえば文字列 “IOI” のアナグラムは、自身を含めて辞書順に “IIO”, “IOI”, “OII” の 3 通りがある。

さて、君にはある文字列が与えられる。君の仕事は、与えられた文字列が、その文字列の全てのアナグラムの中で、辞書順で何番目に現れるか、を調べるプログラムを書くことである。

たとえば、文字列 “EARTH” は “EARTH” のアナグラムの中で 28 番目であり、文字列 “HEART” は “HEART” のアナグラムの中で 55 番目である。

**Input.** 入力ファイル anagram.in から入力を読み込め。入力は 1 行だけであり、この 1 行はアルファベットの大文字のみからなる 20 文字以下の文字列である。この文字列に対して問題の計算を行いなさい。

**Output.** 出力は、標準出力に行うこと。入力で与えられた文字列が、その文字列のアナグラムの中で、辞書順で何番目の文字列であるかを整数で 1 行に出力せよ。

### 例 1

anagram.in	標準出力
HEART	55

### 例 2

anagram.in	標準出力
IOI	2

**注意** オーバーフローに注意すること。

$$2^{32} < 20 \times 19 \times \cdots \times 3 \times 2 \times 1 < 2^{63}$$

である。

## Route: 象使い

input ファイル “route.in”

output 標準出力

ソースファイル route.c/route.cpp/Route.java

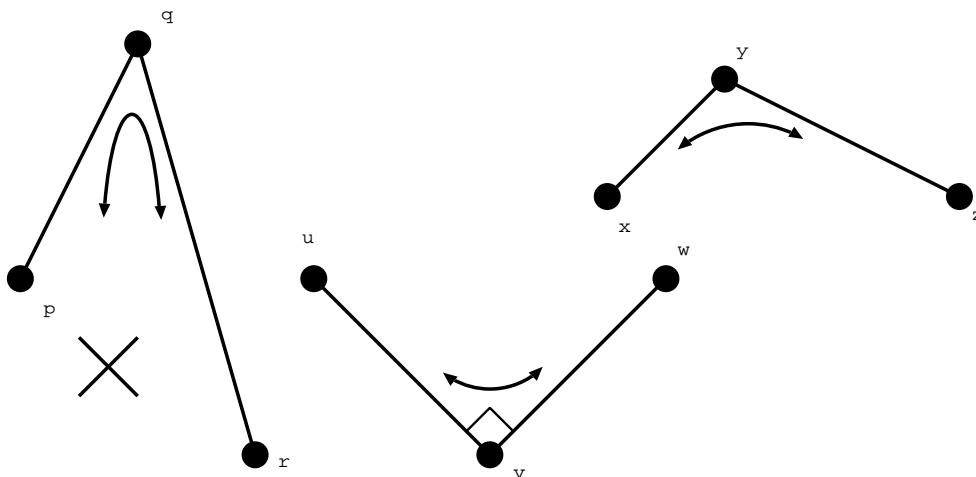
時間制限 1 秒 / データ

あなたの友人の象使いは王宮まで象をつれて行くことを命ぜられた。道路図が与えられるが、王宮までの道路はそれぞれ有料で、さらにその費用は自前で用意しなければならない。

彼のために最も安い道程を探し出して欲しい。

ただし、以下のことに注意すること。

- 道路は2つの料金所の間を結ぶ線分である。2つの料金所  $p, q$  の組に対して、 $p$  と  $q$  を結ぶ道路は高々1本しか存在しない。
- 道路は必ず端点から端点までたどらなければならない、途中でほかの道路に乗り換えることはできない。
- 象は鋭角には曲れないため、料金所では、それまでたどった道路とのなす角が鋭角になる道路には乗り換えられない。例えば、下の図では、 $p \rightarrow q \rightarrow r$  はたどれず、 $u \rightarrow v \rightarrow w$  や  $x \rightarrow y \rightarrow z$  はたどれる。



### Input.

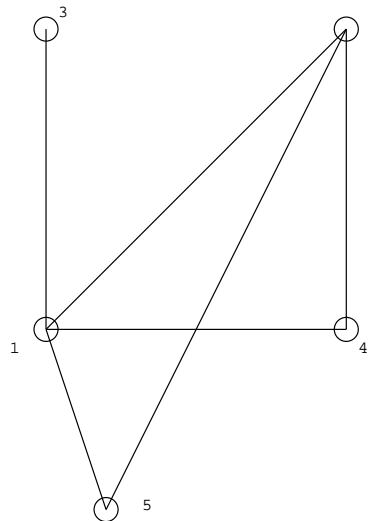
入力ファイル route.in の 1 行目には, 料金所の数  $n$  ( $2 \leq n \leq 100$ ) と道路の数  $m$  が空白で区切られて書かれている.  $i + 1$  行目 ( $1 \leq i \leq n$ ) には, 2 つの整数  $x_i, y_i$  ( $-10000 \leq x_i \leq 10000, -10000 \leq y_i \leq 10000$ ) が空白で区切られて書かれている. これは,  $i$  番目の料金所の座標が  $(x_i, y_i)$  であることを表わしている.  $j + n + 1$  行目 ( $1 \leq j \leq m$ ) には, 3 つの整数  $a_j, b_j, c_j$  ( $1 \leq a_j < b_j \leq n, 0 \leq c_j \leq 10000$ ) が空白で区切られて書かれている. これは,  $a_j$  番目の料金所と  $b_j$  番目の料金所で結ばれる道路の通行料金が  $c_j$  であることを表わしている.

現在の象の居場所は 1 番目の料金所である. 王宮は 2 番目の料金所のすぐ近くにある.

**Output.** 1 番目の料金所から 2 番目の料金所へ到達できる, 最も安い料金を出力せよ. もし到達不可能な場合は,  $-1$  を出力せよ.

### 例

route.in	標準出力
5 6	8
0 0	
10 10	
0 10	
10 0	
2 -6	
1 2 30	
1 3 4	
1 4 5	
1 5 1	
2 4 3	
2 5 1	



入力例の図示

1 番目の料金所から 2 番目の料金所への道程は,  $1 \rightarrow 2$  と  $1 \rightarrow 4 \rightarrow 2$  の 2 つがあり, そのうち料金が安い  $1 \rightarrow 4 \rightarrow 2$  の料金 8 を出力する.  $1 \rightarrow 5 \rightarrow 2$  は, 料金 2 だが, 道路  $1 \leftrightarrow 5$  と道路  $5 \leftrightarrow 2$  が 5 番目の料金所で鋭角をなすので, 象は  $1 \rightarrow 5 \rightarrow 2$  とたどることはできない.

## Circuit: 電気回路の結線

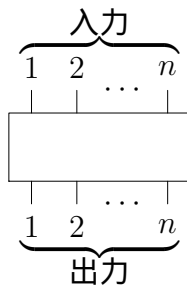
input ファイル “circuit.in”

output 標準出力

ソースファイル circuit.c/circuit.cpp/Circuit.java

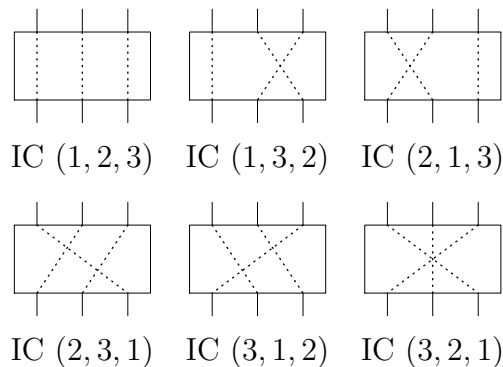
時間制限 1秒 / データ

$n$  個の入力と  $n$  個の出力を持つ, 次の図のような IC (集積回路) を考える.  $n$  個の入力には左から  $1, 2, \dots, n$  と番号が振られている. 同様に  $n$  個の出力にも左から  $1, 2, \dots, n$  と番号が振られている.

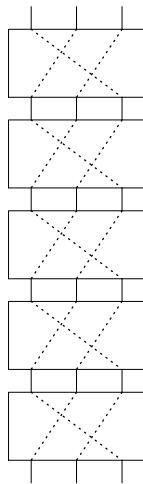


この IC では,  $n$  個の入力の各々がそのまま  $n$  個の出力となって出てくるのだが, どの入力がどの出力に対応しているかはわからない. IC は出力  $1, \dots, n$  に対応する入力の番号を並べることで記述する.

例えば,  $n = 3$  のとき, 次の 6 種類の IC が存在する.



ここで, IC (2, 3, 1) を 5 個直列につなぐと, 次の図のように最初の入力 3, 1, 2 がそれぞれ最後の出力 1, 2, 3 に対応するので, 結局 IC (3, 1, 2) と同じ動作をすることになる.



整数  $n$  ( $1 \leq n \leq 10000$ ),  $k$  ( $1 \leq k \leq 10000$ ) と  $1$  から  $n$  までの異なる整数の列  $a_1, \dots, a_n$  が与えられたとき, 同じ種類の IC を  $k$  個直列につなぐことで IC  $(a_1, \dots, a_n)$  と同じ動作をさせることができるかどうかを判定し, できる場合には目的を達成できる IC を出力するプログラムを書け. 条件に合う IC が複数存在するときは, どの IC を出力してもよい.

#### Input.

入力ファイル `circuit.in` は  $n + 1$  行からなる.

1 行目には整数  $n$  と  $k$  が空白で区切って書かれている.

$i + 1$  行目 ( $1 \leq i \leq n$ ) には整数  $a_i$  が書かれている.

#### Output.

プログラムは結果を標準出力に出力する.

同じ種類の IC を  $k$  個直列につないで IC  $(a_1, \dots, a_n)$  と同じ動作をさせることが可能な場合, 1 行目から  $n$  行目までの  $n$  行で条件を満たす IC を記述せよ. すなわち,  $1 \leq i \leq n$  に対し,  $i$  行目には IC の出力  $i$  に対応する入力の番号を書け.

同じ種類の IC を  $k$  個直列につないで IC  $(a_1, \dots, a_n)$  と同じ動作をさせることが不可能な場合, 0 と書いた 1 行だけを出力せよ.

#### 例 1

circuit.in	標準出力
3 5	2
3	3
1	1
2	

#### 例 2

circuit.in	標準出力
4 4	0
2	
1	
4	
3	