

Bajtuś jest jednym z najmłodszych mieszkańców Bajtogradu. Dopiero niedawno nauczył się pisać i czytać. Jest jednak na tyle duży, że sam już chodzi do szkoły. Codziennie rano wychodzi z domu, a następnie wstępuje po kolei do wszystkich swoich kolegów; dopiero po dołączeniu się ostatniego kolegi cała grupa idzie na lekcje.

Pewnego dnia nauczyciel Bajtusia poprosił go o sporządzenie listy ulic, którymi Bajtuś przechodzi po drodze z domu do szkoły, i odczytanie tej listy na następnych zajęciach. Szybko jednak okazało się, że taka lista mogłaby pochłonąć ogromne ilości papieru. Ustalono więc, że Bajtuś zapisze jedynie pierwszą literę z nazwy każdej ulicy, po której przejdzie. Każda ulica w Bajtogradzie jest jednokierunkowa i łączy dwa różne skrzyżowania.

Bajtuś podczas odczytywania opisu trasy robi przerwy jedynie w miejscach, w których wstępował do kolegów. Możemy więc traktować opis każdego fragmentu jego drogi z domu do szkoły jako jedno słowo. Chłopiec wciąż ma problemy z prawidłowym czytaniem, tj. czytaniem od lewej strony do prawej, i zamiast tego zdarza mu się czytać od strony prawej do lewej. Czasem przeczyta wyraz *mleko* jako *mleko*, a czasem jako *okelm*. Rodzice Bajtusia wiedzą o tych problemach syna, więc postanowili mu pomóc i znaleźć taką trasę, by opis każdego fragmentu, niezależnie od tego, z której strony czytany, brzmiał tak samo. Jednocześnie chcieliby, żeby długość każdego kawałka tej trasy była jak najmniejsza. Zwrócili się do Ciebie z prośbą o pomoc.

Napisz program, który:

- wczyta ze standardowego wejścia opis miasta,
- wyznaczy taką trasę z domu do szkoły, w której każdy fragment jest możliwie najkrótszy, a jego opis, niezależnie od kierunku czytania, brzmi dokładnie tak samo,
- wypisze wyznaczone opisy fragmentów drogi na standardowe wyjście.

Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite n i m oddzielone pojedynczym odstępem ($2 \leq n \leq 400$, $1 \leq m \leq 60\,000$). Oznaczają one odpowiednio liczbę skrzyżowań w Bajtogradzie oraz liczbę łączących je ulic. W kolejnych m wierszach znajdują się opisy ulic. W wierszu $(i + 1)$ -szym znajdują się trzy wartości x_i , y_i , c_i ($1 \leq x_i \leq n$, $1 \leq y_i \leq n$, $x_i \neq y_i$), pooddzielane pojedynczymi odstępami i oznaczające odpowiednio początek ulicy, koniec ulicy oraz pierwszą literę jej nazwy w postaci małej litery alfabetu angielskiego. Między dowolnymi dwoma skrzyżowaniami istnieje maksymalnie jedna ulica w danym kierunku. Kolejny wiersz zawiera jedną liczbę całkowitą d ($2 \leq d \leq 100$), oznaczającą liczbę skrzyżowań, pomiędzy którymi idzie Bajtuś w drodze do szkoły. Następny wiersz zawiera d liczb całkowitych s_i ($1 \leq s_i \leq n$), pooddzielanych pojedynczymi odstępami. Oznaczają one, że Bajtuś mieszka przy skrzyżowaniu numer s_1 , szkoła znajduje się przy skrzyżowaniu s_d , zaś po drodze Bajtuś idzie **kolejno** po kolegów mieszkających przy skrzyżowaniach $s_2, s_3, \dots, s_{d-1}$. Każde dwa następujące po sobie numery skrzyżowań na liście są różne. Może się jednak zdarzyć, że pewne numery skrzyżowań na liście będą takie same. Ponadto, jeśli pomiędzy dwoma kolejnymi skrzyżowaniami nie da się przejść, stosując się do ograniczeń podanych w zadaniu, to Bajtuś idzie na przelaj i niczego nie zapisuje na kartce.

Wyjście

Wyjście powinno składać się z $d - 1$ wierszy. W i -tym wierszu powinna znajdować się liczba r_i oraz ciąg znaków w_i , oddzielone pojedynczym odstępem. Liczba r_i oznacza długość najkrótszej drogi, spełniającej wymogi zadania, łączącej skrzyżowania s_i oraz s_{i+1} , zaś w_i to ciąg pierwszych liter nazw ulic na tej drodze. W przypadku gdy pomiędzy danymi dwoma skrzyżowaniami nie istnieje trasa spełniająca warunki zadania, należy wypisać -1 . Jeśli zaś istnieje kilka możliwych ciągów znaków w_i zgodnych z warunkami zadania, należy wypisać dowolny z nich.

Przykład

Dla danych wejściowych:

```
6 7
1 2 a
1 3 x
1 4 b
2 6 l
3 5 y
4 5 z
6 5 a
3
1 5 3
```

poprawnym wynikiem jest:

```
3 ala
-1
```

