

ICPC ECNA 2016 D - Lost in Translation

Time limit: 0.6s **Memory limit:** 1G

ICPC East Central NA Regional Contest 2016, Problem D

The word is out that you've just finished writing a book entitled *How to Ensure Victory at a Programming Contest* and requests are flying in. Not surprisingly, many of these requests are from foreign countries, and while you are versed in many programming languages, most spoken languages are Greek to you. You've done some investigating and have found several people who can translate between languages, but at various costs. In some cases multiple translations might be needed. For example, if you can't find a person who can translate your book from English to Swedish, but have one person who can translate from English to French and another from French to Swedish, then you're set. While minimizing the total cost of all these translations is important to you, the most important condition is to minimize each target language's distance (in translations) from English, since this cuts down on the errors that typically crop up during any translation. Fortunately, the method to solve this problem is in Chapter 7 of your new book, so you should have no problem in solving this, right?

Input Specification

Input starts with a line containing two integers n m indicating the number of target languages and the number of translators at your disposal ($1 \leq n \leq 100$, $1 \leq m \leq 4\,500$). The second line will contain n strings specifying the n target languages. After this line are m lines of the form l_1 l_2 c where l_1 and l_2 are two different languages and c is a positive integer specifying the cost to translate between them (in either direction). The languages l_1 and l_2 are always either English or one of the target languages, and any pair of languages will appear at most once in the input. The initial book is always written in English.

Output Specification

Display the minimum cost to translate your book to all of the target languages, subject to the constraints described above, or `Impossible` if it is not possible.

Sample Input 1

```
4 6
Pashto French Amheric Swedish
English Pashto 1
English French 1
English Amheric 5
Pashto Amheric 1
Amheric Swedish 5
French Swedish 1
```

Sample Output 1

8

Sample Input 2

2 1
A B
English B 1

Sample Output 2

Impossible