Czech ACM Student Chapter
Czech Technical University in Prague
Faculty of Mathematics and Physics, Charles University in Prague
Faculty of Electrical Engineering and Computer Science, Technical University of Ostrava
Faculty of Informatics and Information Technologies, Slovak University of Technology
Faculty of Informatics, Masaryk University
Faculty of Management Science and Informatics, University of Zilina

CTU Open Contest 2009

Letter Lies

`ll.c`, `ll.C`, `ll.java`, `ll.p`

Some government projects are not done by government offices themselves, but are instead contracted to private companies. The selection of the best candidate is a task for a specially selected experts. Before the selection process starts, all the candidates send letters with their offer. The committee then picks the best offer based on its price and quality.

That was the theory... In reality, the selection process is often just a formality and the winner was chosen long before, based on the bribe given to the committee. To masquerade these frauds, fake candidate companies are included in the process. And, of course, it is necessary to pretend that the offer letters by these companies have been received. Some people are involved in such selection procedures so often that they implemented an automated letter generator to make their work easier. The generator can assemble letters from a list of pre-defined sentences and the following rules:

- Some sentences are greetings and can appear only at the beginning of the letter. Each letter has to start with a greeting.
- Some sentences are closings and can appear only at the end. Each letter has to end with a closing.
- No sentence can appear twice in the same letter.
- Successor rules: Each sentence restricts the list of other sentences that can follow. For example, a sentence saying "`Hello`" cannot be followed by "`This concludes our offer`".
- The resulting letter must have an appropriate length. Therefore, the exact number of sentences is specified.

Would you help ACM to detect fake letters generated by this program? You are given a set of letter-generator rules and your task is to compute the total number of different valid letters that can be assembled.

Input Specification

The input will consist of several test scenarios. Each scenario starts by a line with four positive integers N, L, B, F. N is the number of all sentences ($1 \leq N \leq 1000$), B and F are the number of greetings and closings ($1 \leq B, F \leq 1000$), and L is the desired length ($1 \leq L \leq 1000000$).

Then there are N lines with successor rules; i-th of them is composed of the number i, integer D_i ($0 \leq D_i \leq 1000$), and a list of D_i integers determining the sentences that can follow sentence number i.

Next B lines contain numbers of all possible starting sentences and the last F lines numbers of all possible final sentences.

The last scenario is followed by a line containing four zeros.

The author of the generator did not implement any check that some sentence occurs more than once. Instead, this is always guaranteed by the set of successor rules. If these rules are followed, you may safely assume that no letter can ever contain the same sentence twice, that a greeting can appear only at the beginning and a closing only at the end of the letter.

Output Specification

For each scenario, output a line with one integer number — the total number of valid letters with exactly L sentences. Please note that this number may exceed 2^{32}. If there is no valid letter of the given length, output the string "`impossible`".

Sample Input

```
4 2 2 2
1 1 3
2 1 4
3 0
4 0
1
2
4
3
2 1000000 1 1
1 1 2
2 0
1
2
0 0 0 0
```

Output for Sample Input

```
2
impossible
```