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

Odd Opportunities

`oo.c`, `oo.C`, `oo.java`, `oo.p`

ACM discovered there is a secret worldwide organization (Mafia) that operates in complete secrecy. Each member of Mafia has contacts to some other members, but not necessarily to all of them.

The new (recently elected) head of the Mafia came with a brilliant plan to make their operation even more secret: Some members will have to eliminate some of their contacts and to stop communicating with them. It has been announced that it is very important whether the number of contacts is odd or even. Each member has been instructed whether the number of their contacts must remain an odd or an even number.

A little bit of chaos developed after this announcement. Nobody understands why this is so important, but nobody dares to protest. Your task is to write a program that will suggest a suitable solution and select contacts to be dropped.

Input Specification

The input will consist of several test scenarios. Each scenario starts by a line with two positive integers V and E. V is the number of Mafia members ($1 \leq V \leq 1000$) and E is the total number of contacts ($0 \leq E \leq V.(V-1)/2$).

Then there are E lines describing individual contacts. Each line contains two integers v_1 and v_2, which means that there is a contact between members v_1 and v_2 ($1 \leq v_1, v_2 \leq N$, $v_1 \neq v_2$).

Then there is one more line in each scenario containing exactly V lowercase characters. Each character corresponds to one member and is either "`o`" (the number of contacts must remain odd) or "`e`" (the number of contacts must remain even). Obviously, the first character corresponds to member 1, second character to member 2, etc.

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

Output Specification

For each scenario, output a subset of original contacts such that will satisfy the even/odd conditions for all members. On the first line, output one integer number R — the number of contacts that remain ($0 \leq R \leq E$). Then, output R lines, each of them specifying two members that remain in contact. Make sure no pair appears more than once.

If it is not possible to find a suitable subset of contacts, output the word "`impossible`". If there are more correct solutions, you may choose any of them.

Sample Input

```
5 6
1 2
2 3
3 4
4 5
1 3
1 4
oeooo
3 1
1 2
oeo
5 0
eeeee
5 0
eeoee
5 0
eeoeo
4 2
1 2
4 3
eeee
0 0
```

Output for Sample Input

```
3
4 5
4 3
1 4
impossible
0
impossible
impossible
0
```