

ЗАДАЧА 3

Консултация

Една консултанска фирма е организирана работата си по следния начин: Всички клиенти чакат на опашка и консултациите са в съответствие с реда в опашката – първият дошъл се обслужва пръв. Консултантът за даден клиент се определя според пола на предишния клиент – мъж (М) или жена (F). Например, консултантът C_i ($i=1,2,\dots,N$), ($2 < N \leq 40$), който приема клиент, консултира клиента и след това насочва следващия клиент към консултант C_j , ако вече обработеният клиент е бил мъж, или към консултант C_k , ако вече обработеният клиент е бил жена. Тогава, този нов консултант обслужва следващия клиент.

Един умен клиент забелязал, че понякога е възможно той да бъде на такова място в опашката, че независимо от консултанта, който започва консултациите, този умен клиент да бъде обслужен от точно определен консултант C_m .

Напишете програма, която намира дали това е възможно. Ако е възможно, програмата трябва да намери опашка с минимална дължина (числото изразяващо тази дължина трябва да дава броя на клиентите преди умния клиент), завършваща с умния клиент и определя консултанта, който ще обслужи този клиент. В противен случай, ако такава опашка не съществува, програмата трябва да запише в изходния файл съобщението 'Impossible'.

Консултантите са номерирани с целите числа от 1 до N .

Входен файл

Първият ред на входния файл `c:\day1\problem3\visit.in` съдържа числото N , ($2 < N \leq 40$), равно на броя на консултантите от фирмата.

В следващите N реда има по 3 числа. Първото число е номер на консултант, второто е номера на следващия консултант, когато консултираният клиент е бил жена, а третото число е номера на следващия консултант, когато консултираният клиент е бил мъж.

Изходен файл

Изходният файл `c:\day1\problem3\visit.out` трябва да съдържа 3 реда, в случай че търсената опашка съществува или само един ред, в противен случай.

Ако търсената опашка съществува, първият ред на файла трябва да съдържа броя на лицата в опашката (без да се брои самият умен клиент), вторият ред трябва да съдържа низ от букви F, M, който представя опашката, и третият ред трябва да съдържа номера на консултанта, който ще консултира умния клиент.

Ако търсената опашка не съществува, изходният файл трябва да съдържа низа 'Impossible'.

Изпълним файл `c:\day1\problem3\visit.exe`.

Време за изпълнение на всеки тест 5 секунди.

Пример 1

Входен файл:

visit.in

```
5
1      2      4
2      4      1
3      2      3
4      1      4
5      3      5
```

Изходен файл

visit.out

```
4
FFMM
4
```

Пример 2.

Входен файл

visit.in

```
4
1      4      2
2      3      1
3      1      4
4      2      3
```

Изходен файл

visit.out

Impossible