

## Задача А. Обход в ширину

Имя входного файла: `bfs.in`  
Имя выходного файла: `bfs.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. В нём необходимо найти расстояние от одной заданной вершины до другой.

### Формат входного файла

В первой строке входного файла содержится три натуральных числа  $N$ ,  $S$  и  $F$  ( $1 \leq S, F \leq N \leq 100$ ) — количество вершин в графе и номера начальной и конечной вершин соответственно. Далее в  $N$  строках задана матрица смежности графа. Если значение в  $j$ -м элементе  $i$ -й строки равно 1, то в графе есть направленное ребро из вершины  $i$  в вершину  $j$ .

### Формат выходного файла

В единственной строке должно находиться минимальное расстояние от начальной вершины до конечной. Если пути не существует, выведите 0.

### Пример

bfs.in	bfs.out
4 4 3 0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0	2

## Задача В. Кратчайший путь коня

Имя входного файла: `knight1.in`  
Имя выходного файла: `knight1.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

На шахматной доске размером  $8 \times 8$  заданы две клетки. Соедините эти клетки кратчайшим путем коня.

### Формат входного файла

Программа получает на вход координаты двух клеток, каждая в отдельной строке. Координаты клеток задаются в виде буквы (от “a” до “h”) и цифры (от 1 до 8) без пробелов.

### Формат выходного файла

Программа должна вывести путь коня, начинающийся и заканчивающийся в данных клетках и содержащий наименьшее число клеток.

### Пример

knight1.in	knight1.out
a1 h8	a1 b3 a5 b7 d8 f7 h8

## Задача С. Кратчайший путь двух коней

Имя входного файла: `knight2.in`  
Имя выходного файла: `knight2.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайта

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке.

### Формат входного файла

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

### Формат выходного файла

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений.

### Пример

knight2.in	knight2.out
a1	1 b3
c2	1 d4
c2	2 a1
a1	1 c2

## Задача D. Островные государства

Имя входного файла: island2.in  
Имя выходного файла: island2.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль. Кроме этого за выезд из городов с четными номерами берется удвоенная пошлина.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город  $N$  на Всебайтландское сборище программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

### Формат входного файла

В первой строке входного файла записано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100\,000$ ) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах —  $N$  чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних  $M$  строках записаны пары  $1 \leq a, b \leq N$ ,  $a \neq b$ . Каждая пара означает наличие дороги из города  $a$  в город  $b$ . По дорогам Байтландии можно двигаться в любом направлении.

### Формат выходного файла

Если искомого пути не существует, выведите единственное слово **impossible**. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

### Пример

island2.in	island2.out
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 1 1 2 1 1 2 2 3 3 4 4 5 2 4	3 5 1 2 3 4 5

## Задача Е. Расстояние от корня

Имя входного файла: `rootdist.in`  
Имя выходного файла: `rootdist.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В заданном корневом дереве найдите вершины, максимально удалённые от корня. Расстоянием между вершинами считается количество рёбер в пути.

### Формат входного файла

В первой строке задано  $n$  — количество вершин в дереве ( $1 \leq n \leq 100$ ). В следующих  $n - 1$  строках заданы вершины, являющиеся предками вершин  $2, 3, \dots, n$ . Вершина 1 является корнем дерева.

### Формат выходного файла

В первой строке выведите максимальное расстояние от корня до остальных вершин дерева. Во второй строке выведите, сколько вершин дерева находятся от корня на таком расстоянии. В третьей строке выведите номера этих вершин через пробел в порядке возрастания.

### Примеры

<code>rootdist.in</code>	<code>rootdist.out</code>
3	1
1	2
1	2 3
3	2
1	1
2	3

## Задача F. Почтальон

Имя входного файла: `post.in`  
Имя выходного файла: `post.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В городе  $\mathfrak{C}$  есть  $n$  площадей, соединенных улицами. При этом количество улиц не превышает ста тысяч и существует не более трех площадей, на которые выходит нечетное число улиц. Для каждой улицы известна ее длина. По улицам разрешено движение в обе стороны. В городе есть хотя бы одна улица. От любой площади до любой можно дойти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице так, что бы длина его пути была наименьшей. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество площадей в городе ( $1 \leq n \leq 1000$ ). Далее следуют  $n$  строк, задающих улицы. В  $i$ -ой из этих строк находится число  $m_i$  — количество улиц, выходящих из площади  $i$ . Далее следуют  $m_i$  пар положительных чисел. В  $j$ -ой паре первое число — номер площади, в которую идет  $j$ -ая улица с  $i$ -ой площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улиц с площади на нее саму.

Все числа во входном файле не превосходят  $10^5$ .

### Формат выходного файла

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте (считая первую и последнюю), а во вторую — номера площадей в порядке их посещения.

Если решений нет, выведите в выходной файл одно число -1.

Если решений несколько, выведите любое.

### Пример

<code>post.in</code>	<code>post.out</code>
4	5
2 2 1 2 2	1 2 3 4 2 1
4 1 2 4 4 3 5 1 1	
2 2 5 4 8	
2 3 8 2 4	