

## Содержание

<b>Must have</b>	<b>2</b>
Задача 15А. Любители Кошек [1, 256]	2
Задача 15В. 2-SAT [1, 256]	3
<b>Обязательные задачи</b>	<b>4</b>
Задача 15С. Points. Точки сочленения [1, 256]	4
Задача 15D. Компоненты вершинной двусвязности [1, 256]	5
Задача 15Е. Союзы [1, 256]	6
Задача 15F. Раскраска в три цвета [1, 256]	7
<b>Дополнительные задачи</b>	<b>8</b>
Задача 15G. Союз-2 [1, 256]	8
Задача 15H. Chip Installation [1, 256]	9

---

У вас не получается читать/выводить данные?  
Воспользуйтесь примерами (**c++**) (**python**).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`.

Подни можно пользоваться **дополнительной библиотекой** (`optimization.h`).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет `vector-set-map-весь-STL`): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

## Must have

### Задача 15А. Любители Кошек [1, 256]

В университетском клубе любителей кошек зарегистрировано  $n$  членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

#### Формат входных данных

В первой строке входного файла заданы числа  $n$  и  $m$  ( $1 \leq n \leq 1000$ ,  $1 \leq m \leq 30000$ ), где  $m$  обозначает общее число знакомств. В последующих  $m$  строках идут пары чисел  $a_i$   $b_i$ , обозначающие, что  $a_i$  знаком с  $b_i$ . Информация об одном знакомстве может быть записана несколько раз, причем даже в разном порядке (как  $(x, y)$ , так и  $(y, x)$ ).

#### Формат выходных данных

В выходной файл необходимо вывести количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

#### Пример

stdin	stdout
3 3 1 2 2 3 3 1	1

#### Замечание

Предполагается решение за  $\mathcal{O}(nm)$ .

Если каждый треугольник сразу считать ровно один раз, получится быстрее.

### Задача 15B. 2-SAT [1, 256]

Широко известна задача **2-SAT**. Решите её. Гарантируется, что решение существует.

Формулировка 2-SAT: нужно подобрать значения  $n$  булевых переменных так, чтобы все  $m$  утверждений вида  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  обратились в истину.

#### Формат входных данных

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных  $n$  и число утверждений  $m$ . Каждая из следующих  $m$  строк содержит числа  $i_1, e_1, i_2, e_2$ , задает утверждение  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  ( $0 \leq i_j < n, 0 \leq e_j \leq 1$ ).

Ограничения: сумма всех  $n$  не больше 100 000, сумма всех  $m$  не больше 300 000.

#### Формат выходных данных

Для каждого теста выведите строку из  $n$  нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

#### Примеры

stdin	stdout
1 0	1
2 2	01
0 0 1 0	000
0 1 1 1	
3 4	
0 1 1 0	
0 0 2 1	
1 1 2 0	
0 0 0 1	

## Обязательные задачи

### Задача 15С. Points. Точки сочленения [1, 256]

Дан неориентированный граф без петель а кратных рёбер. Требуется найти все точки сочленения в нем.

#### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $n \leq 20\,000$ ,  $m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

#### Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество точек сочленения в заданном графе. На следующей строке выведите  $b$  целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

#### Пример

stdin	stdout
9 12	3
1 2	1
2 3	2
4 5	3
2 6	
2 7	
8 9	
1 3	
1 4	
1 5	
6 7	
3 8	
3 9	

#### Замечание

По мотивам лекции.

### Задача 15D. Компоненты вершинной двусвязности [1, 256]

Компонентой вершинной двусвязности графа  $\langle V, E \rangle$  называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

#### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и ребер графа соответственно ( $n \leq 20\,000$ ,  $m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

#### Формат выходных данных

В первой строке выходного файла выведите целое число  $k$  — количество компонент вершинной двусвязности графа. Во второй строке выведите  $m$  натуральных чисел  $a_1, a_2, \dots, a_m$ , не превосходящих  $k$ , где  $a_i$  — номер компоненты вершинной двусвязности, которой принадлежит  $i$ -е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

#### Примеры

stdin	stdout
5 6	2
1 2	1 1 1 2 2 2
2 3	
3 1	
1 4	
4 5	
5 1	

#### Замечание

По мотивам лекции. Сдайте сперва предыдущую задачу.

## Задача 15Е. Союзы [1, 256]

### Легенда

В Триаметистовом королевстве было  $n$  городов и  $m$  дорог, соединявших некоторые из них друг с другом. Однажды в результате экспериментов придворного мага Д произошло катастрофическое расщепление: во вселенной вместо одного Триаметистового королевства появилось множество его копий, или *отражений*. Более того, в каждом из них каждая дорога стала заколдованной либо способом  $\alpha$ , либо способом  $\omega$  (стоит отметить, что каждое из возможных сочетаний заколдованностей дорог появилось ровно в одном из отражений).

Свойства заколдованностей  $\alpha$  и  $\omega$  таковы, что города  $a$ ,  $b$  и  $c$  образуют торговый союз тогда и только тогда, когда есть дороги между  $a$  и  $b$ , между  $b$  и  $c$  и между  $c$  и  $a$ , заколдованные одним и тем же способом.

### Задача

Найти число треугольников в неорграфе.

### Формат входных данных

В первой строке входных данных записаны два целых числа  $n$  и  $m$  — количество городов и дорог Триаметистового королевства. В следующих  $m$  строках записаны пары чисел, задающие города, соединённые соответствующими дорогами.  $3 \leq n \leq 10\,000$ ,  $1 \leq m \leq 100\,000$ . Никакие два города не соединены более чем одной дорогой, никакая дорога не соединяет город сам с собой.

### Формат выходных данных

Выведите как можно точнее единственное вещественное число — среднее количество союзов, которое образовалось во всех отражениях Триаметистового королевства.

### Пример

stdin	stdout
3 3 1 2 2 3 3 1	0.25

### Задача 15F. Раскраска в три цвета [1, 256]

Петя нарисовал на бумаге  $n$  кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

#### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  — количество кружков и количество линий, которые нарисовал Петя, соответственно ( $1 \leq n \leq 1\,000$ ,  $0 \leq m \leq 20\,000$ ).

Следующая строка содержит  $n$  символов из множества {"R", "G", "B"} —  $i$ -й из этих символов означает цвет, в который раскрашен  $i$ -й кружок ("R" — красный, "G" — зеленый, "B" — синий).

Следующие  $m$  строк содержат по два целых числа — пары кружков, соединенных отрезками.

#### Формат выходных данных

Выведите в выходной файл одну строку, состоящую из  $n$  символов из множества {"R", "G", "B"} — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово "Impossible".

#### Пример

stdin	stdout
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	BBGR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

#### Замечание

Задача с практики. Можно сдать за  $O(nm)$ , можно за  $O(n + m)$ .

## Дополнительные задачи

### Задача 15G. Союз-2 [1, 256]

#### Легенда

В Триаметистовом королевстве было  $n$  городов и  $m$  дорог, соединявших некоторые из них друг с другом. Однажды в результате экспериментов придворного мага Д произошло катастрофическое расщепление: во вселенной вместо одного Триаметистового королевства появилось множество его копий, или *отражений*. Более того, в каждом из них каждая дорога стала заколдованной либо способом  $\alpha$ , либо способом  $\omega$  (стоит отметить, что каждое из возможных сочетаний заколдованностей дорог появилось ровно в одном из отражений).

Свойства заколдованностей  $\alpha$  и  $\omega$  таковы, что города  $a$ ,  $b$  и  $c$  образуют торговый союз тогда и только тогда, когда есть дороги между  $a$  и  $b$ , между  $b$  и  $c$  и между  $c$  и  $a$ , заколдованные одним и тем же способом.

#### Задача

Найти число треугольников в неорграфе.

#### Формат входных данных

В первой строке входных данных записаны два целых числа  $n$  и  $m$  — количество городов и дорог Триаметистового королевства. В следующих  $m$  строках записаны пары чисел, задающие города, соединённые соответствующими дорогами.  $3 \leq n \leq 10^6$ ,  $1 \leq m \leq 2 \cdot 10^6$ . Никакие два города не соединены более чем одной дорогой, никакая дорога не соединяет город сам с собой.

#### Формат выходных данных

Выведите как можно точнее единственное вещественное число — среднее количество союзов, которое образовалось во всех отражениях Триаметистового королевства.

#### Пример

stdin	stdout
3 3 1 2 2 3 3 1	0.25

#### Замечание

Задача отличается от предыдущей только ограничениями.

### Задача 15Н. Chip Installation [1, 256]

Новый ЧИП скоро установят в новый летательный аппарат, недавно выпущенной компанией Airtram. ЧИП имеет форму диска. Есть  $n$  проводов, которые нужно подсоединить к ЧИПу.

Каждый провод можно подсоединить в один из двух разъемов, допустимых для этого провода. Все  $2n$  разъемов расположены на границе диска. По кругу. Каждый провод имеет свой цвет. Для повышения безопасности два провода одного цвета не могут быть подсоединены к соседним разъемам.

Дана конфигурация разъемов на ЧИПе, найдите способ подсоединить все провода, не нарушающий условия про цвета.

#### Формат входных данных

Первая строка содержит число  $n$  — количество проводов ( $1 \leq n \leq 50\,000$ ). Вторая строка содержит  $n$  целых чисел от 1 до  $10^9$  — цвета проводов. Цвета проводов могут совпадать. Третья строка содержит  $2n$  целых чисел от 1 до  $n$  описывающих разъемы. Число обозначает номер провода, который может быть подсоединен к данному разъему. Каждое число от 1 до  $n$  встречается ровно дважды. Разъемы перечислены в порядке "по кругу". 1-й разъем является соседним со 2-м и так далее, не забудьте, что  $n$ -й является соседним с 1-м.

#### Формат выходных данных

Если не существует способа подключить все провода, выведите одно слово "NO".

Иначе выведите "YES" и  $n$  целых чисел. Для каждого провода выведите номер разъема, к которому нужно подключить этот провод. Разъемы нумеруются числами от 1 до  $2n$  в том порядке, в котором они даны во входном файле.

#### Пример

stdin	stdout
2 1 1 1 1 2 2	YES 1 3
2 1 1 1 2 1 2	NO
2 1 2 1 2 1 2	YES 1 2

#### Замечание

Две задачи по цене одной: эта есть и в теордз.