

## Содержание

Разные задачи	2
1 Задача А. Сочетания	2
2 Задача В. Культурги и скобки	3
3 Задача С. Квадратные корни	4
4 Задача D. Шифровка	5
Графы	6
5 Задача Е. От списка ребер к матрице смежности	6
6 Задача F. Светофоры	7
7 Задача G. От матрицы смежности к списку ребер	8

## Разные задачи

### 1 Задача А. Сочетания

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

Во входном файле заданы числа  $n$  и  $k$ . Выведите в выходной файл все сочетания по  $k$  из чисел от 1 до  $n$  в лексикографическом порядке.  $1 \leq k \leq n \leq 16$ .

#### Пример

<code>choose.in</code>	<code>choose.out</code>
4 2	1 2 1 3 1 4 2 3 2 4 3 4

## 2 Задача В. Культорги и скобки

Имя входного файла: `brackets.in`  
Имя выходного файла: `brackets.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Встретились однажды три культорга ЛКШ ...

- Первый культорг написала мелом на доске скобочную последовательность.
- Второму культоргу стало интересно, существует ли циклический сдвиг, превращающий эту последовательность в правильную.
- Третий же культорг, немного подумав, сказал, сколько таких сдвигов существует.

Вам известна скобочная последовательность, записанная первым культоргом. Найдите число, которое произнёс третий культорг.

Циклическим сдвигом строки называется перенос некоторого (возможно, нулевого) количества символов из конца строки в её начало без изменения их порядка.

### Формат входных данных

В единственной строке дана скобочная последовательность, записанная первым культоргом. Длина последовательности не равна нулю и не превышает 100000 символов.

### Формат выходных данных

Выведите количество циклических сдвигов, превращающих записанную скобочную последовательность в правильную.

### Примеры

<code>brackets.in</code>	<code>brackets.out</code>
<code>)((</code>	1
<code>)()( (</code>	2
<code>()</code>	1

### 3 Задача С. Квадратные корни

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

Ваша задача — найти вещественные корни уравнения  $ax^2 + bx + c = 0$ .

#### Формат входных данных

Входной файл содержит три целых числа  $a$ ,  $b$  and  $c$ . Они не могут превышать  $10^9$  по модулю.

#### Формат выходных данных

В первой строке выходного файла выведите число корней уравнения. Следующие  $k$  строк содержат по одному вещественному числу, округленному до шести знаков после запятой, — корни этого уравнения. Корни должны располагаться в порядке возрастания. Если корней бесконечно много, выведите одно числ  $-1$  вместо числа корней.

#### Пример

<code>sroots.in</code>	<code>sroots.out</code>
1 -2 -3	2 -1.000000 3.000000

## 4 Задача D. Шифровка

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

Мюллер много раз пытался поймать Штирлица с поличным, но тот всё время выкручивался. Как-то раз Штирлиц просматривал электронную почту. В это время незаметно вошел Мюллер и увидел, как у него на экране появился бессмысленный набор символов.

“Шифровка”, — подумал Мюллер.

“UTF-8”, — подумал Штирлиц.

Известно, что Штирлиц шифрует текст простым алгоритмом: он многократно вставляет в произвольное место текста две одинаковые буквы. Вы должны восстановить исходный текст.

### Формат входных данных

В единственной строке записана шифровка Штирлица, состоящая из строчных латинских букв. Длина шифровки не превосходит 200000.

### Формат выходных данных

Выведите восстановленный текст.

### Пример

<code>decode.in</code>	<code>decode.out</code>
<code>wwstdaadierfflitzzz</code>	<code>stierlitz</code>

## Графы

### 5 Задача E. От списка ребер к матрице смежности

Имя входного файла: e2m.in  
Имя выходного файла: e2m.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Простой неориентированный граф задан списком ребер, выведите его представление в виде матрицы смежности.

#### Формат входных данных

Входной файл содержит числа  $N$  ( $1 \leq N \leq 100$ ) — число вершин в графе и  $M$  ( $1 \leq M \leq \frac{n(n-1)}{2}$ ) — число ребер. Затем следует  $M$  пар чисел — ребра графа.

#### Формат выходных данных

Выведите в выходной файл матрицу смежности заданного графа.

#### Пример

e2m.in	e2m.out
3 3	0 1 1
1 2	1 0 1
2 3	1 1 0
1 3	

## 6 Задача F. Светофоры

Имя входного файла: `lights.in`  
Имя выходного файла: `lights.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайт

В подземелье  $M$  тоннелей и  $N$  перекрестков, каждый тоннель соединяет какие-то два перекрестка. Мышиный король решил поставить по светофору в каждом тоннеле перед каждым перекрестком. Напишите программу, которая посчитает, сколько светофоров должно быть установлено на каждом из перекрестков. Перекрестки пронумерованы числами от 1 до  $N$ .

### Формат входных данных

Во входном файле записано два числа  $N$  и  $M$  ( $0 < N \leq 100$ ),  $0 \leq M \leq \frac{N(N-1)}{2}$ ). В следующих  $M$  строках записаны по два числа  $i$  и  $j$  ( $1 \leq i, j \leq N$ ), которые означают, что перекрестки  $i$  и  $j$  соединены тоннелем.

### Формат выходных данных

В выходной файл вывести  $N$  чисел:  $k$ -е число означает количество светофоров на  $k$ -м перекрестке.

### Пример

<code>lights.in</code>	<code>lights.out</code>
7 10 5 1 3 2 7 1 5 2 7 4 6 5 6 4 7 5 2 1 5 3	3 3 2 2 5 2 3

## 7 Задача G. От матрицы смежности к списку ребер

Имя входного файла: `m2e.in`  
Имя выходного файла: `m2e.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Простой неориентированный граф задан матрицей смежности, выведите его представление в виде списка ребер.

### Формат входных данных

Входной файл содержит число  $N$  ( $1 \leq N \leq 100$ ) — число вершин в графе, и затем  $N$  строк по  $N$  чисел, каждое из которых равно 0 или 1 — его матрицу смежности.

### Формат выходных данных

Выведите в выходной файл список ребер заданного графа. Ребра можно выводить в произвольном порядке.

### Пример

<code>m2e.in</code>	<code>m2e.out</code>
3	1 2
0 1 1	2 3
1 0 1	1 3
1 1 0	