

Условие  
недоступно  
на  
русском  
языке

## 1В. Покрывтие путями

2 секунды, 64 мегабайта

{64 мегабайта}

Задан ориентированный ациклический граф. Требуется определить минимальное количество не пересекающихся по вершинам путей, покрывающих все вершины.

### Входные данные

Первая строка входного файла содержит целые числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $2 \leq n \leq 1\,000$ ,  $0 \leq m \leq 10^5$ ). В следующих  $m$  строках содержатся по два натуральных числа — номера вершин  $u$  и  $v$ , которые соединяет ребро  $(u, v)$ .

### Выходные данные

В первой строке выходного файла выведите натуральное число  $k$  — минимальное количество путей, необходимых для покрытия всех вершин.

### Входные данные

3 3  
1 3  
3 2  
1 2

### Выходные данные

1

## 1С. Замощение доминошками

2 секунды, 64 мегабайта

Дано игровое поле размера  $n \times m$ , некоторые клетки которого уже замощены. Замостить свободные соседние клетки поля доминошкой размера  $1 \times 2$  стоит  $a$  условных единиц, а замостить свободную клетку поля квадратиком размера  $1 \times 1$  —  $b$  условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить всё поле.

### Входные данные

Первая строка входного файла содержит 4 целых числа  $n, m, a, b$  ( $1 \leq n, m \leq 100$ ,  $|a| \leq 1\,000$ ,  $|b| \leq 1\,000$ ). Каждая из последующих  $n$  строк содержит по  $m$  символов: символ "." (точка) обозначает занятую клетку поля, а символ "\*" (звёздочка) — свободную.

### Выходные данные

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно замостить свободные клетки поля (и только их).

входные данные
2 3 3 2 .*.* .*.*
выходные данные
5

## 1D. Такси

0.5 секунд, 256 мегабайт

Управлять службой такси — совсем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел:  $x$ -координатой и  $y$ -координатой. Время, необходимое для того, чтобы добраться из точки с адресом  $(a, b)$  в точку  $(c, d)$ , равно  $|a - c| + |b - d|$  минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

### Входные данные

В первой строке входного файла записано число заказов  $M$  ( $0 < M < 500$ ). Последующие  $M$  строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате  $hh:mm$  (в интервале с  $00:00$  по  $23:59$ ), координаты  $(a, b)$  точки отправления и координаты  $(c, d)$  точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

### Выходные данные

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

входные данные
2 08:00 10 11 9 16 08:07 9 16 10 11
выходные данные
1

входные данные
2 08:00 10 11 9 16 08:06 9 16 10 11
выходные данные
2

## 1E. Фруктовый сок

1 секунда, 256 мегабайт

Степан недавно купил себе новую соковыжималку. Теперь по утрам он и его братья и сестры пьют свежавыжатый фруктовый сок. А это, между прочим, очень полезно!

Недавно они поняли, что можно пить сок, выжатый не только из одного вида фруктов, как, например, апельсиновый, но и различные смеси, например, виноградно-яблочный.

В семье Степана все очень любят сок, поэтому могут утром выпить не один стакан, причем разных видов сока. Например, его сестра Катя очень любит грейпфрутовый и апельсиновый соки. Степан, как наиболее технически грамотный человек, каждое утро занимается приготовлением соков.

Опишем подробнее, как работает соковыжималка. В нее загружаются фрукты, они проходят отжим в центрифуге, обезвоженная мякоть сбрасывается в отдельный резервуар, а сок попадает в специальную емкость.

Основная проблема состоит в том, что эту емкость иногда приходится мыть. Например, если после приготовления апельсинового сока, необходимо приготовить яблочный, то емкость надо мыть, иначе получится апельсиново-яблочный сок. Более формально, пусть сок  $A$  состоит из компонентов  $a_1, \dots, a_n$ , а сок  $B$  – из компонентов  $b_1, \dots, b_m$ . Сок  $B$  можно готовить после сока  $A$ , если любой из компонентов  $a_i$  является компонентом сока  $B$  (т.е.  $\exists j : b_j = a_i$ ). В противном случае емкость для сока надо помыть.

Степан не очень любит мыть посуду, поэтому хочет мыть емкость как можно меньшее число раз. Помогите ему.

**Входные данные**

Первая строка входного файла содержит количество  $N$  различных соков, которые требуется приготовить ( $1 \leq N \leq 300$ ). Каждая из последующих  $N$  строк описывает один из соков. Описание сока состоит из числа  $k$  его компонентов ( $1 \leq k \leq 300$ ) и списка этих компонентов. Каждый из компонентов сока описывается словом длиной до 30 символов из строчных и прописных букв латинского алфавита. Прописные и строчные буквы различаются. Различные компоненты имеют различные названия.

**Выходные данные**

В выходной файл выведите минимальное количество раз, которое Степану придется помыть емкость для сока. Учитывайте при этом, что емкость для сока надо помыть и после приготовления последней порции сока.

Входные данные
4 1 Apple 2 Apple Orange 1 Orange 2 Orange Pineapple
Выходные данные
2

Входные данные
3 1 Apple 1 Orange 1 Mango
Выходные данные
3

1F. 3a Орду!

2 секунды, 256 мегабайт

Сегодня великий вождь Саурфанг поведет орков в бой, и орки планируют сражаться так яростно, как никогда прежде. Любый великий вождь знает, что для успешной атаки нужно тщательное планирование.

В бою будут участвовать  $n$  орков, которых необходимо разделить на отряды. Саурфанг считает, что важны не количество или численность отрядов, а высокий боевой дух. Для каждого орка известно, что сам он пойдет в атаку с боевым духом  $a_i$ . Однако если орка воодушевит его командир, то он уже будет сражаться более яростно с боевым духом  $b_i$ . Но здесь возникает небольшая сложность...

Система подчинения орков весьма сложна. У каждого орка может быть несколько командиров, а у каждого командира может быть много подчиненных. Известно лишь, что все орки разного возраста, и никогда младший орк не может командовать старшим.

Саурфанг решил, что каждый отряд будет устроен следующим образом: вождь назначит командира отряда, который пойдет впереди. За ним будет идти его подчиненный, следующим будет подчиненный второго орка, и так далее. При этом воодушевлены будут все орки, кроме командира отряда. Возможно, что отряд будет состоять лишь из одного невоодушевленного орка.

Осталось лишь разделить орков на отряды, так, чтобы каждый орк был ровно в одном отряде, а суммарный боевой дух орков был максимальным. Но поскольку Саурфанг — орк, а орки не любят решать задачи, он пойдет точить свой топор, а сформировать отряды предстоит вам.

**Входные данные**

В первой строке задано число орков  $n$  ( $1 \leq n \leq 10^5$ ).

Далее идет  $n$  строк. В  $i$ -й строке задано число  $k_i$  — количество орков, подчиняющихся  $i$ -му, а затем  $k_i$  чисел — номера этих орков. Орки пронумерованы в порядке старшинства, поэтому каждое число в  $i$ -й строке строго больше  $i$ . Сумма  $k_i$  не превосходит  $2 \cdot 10^5$ .

В следующей строке содержатся  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ).

В последней строке содержатся  $n$  целых чисел  $b_i$  ( $a_i \leq b_i \leq 10^9$ ).

**Выходные данные**

В первой строке выведите два числа — суммарный боевой дух орков при оптимальном разбиении на отряды, и количество отрядов  $s$ .

В следующих  $s$  строках выведите описания отрядов. В начале описания выведите количество орков в отряде  $t_j$ , а затем  $t_j$  чисел — номера орков по порядку, начиная с командира отряда.

Если оптимальных разбиений существует несколько, выберите любое.

входные данные
4 2 2 3 1 4 1 4 0 1 1 1 1 1 2 3 4
выходные данные
9 2 2 1 3 2 2 4

1G. Смешной граф

1 секунда, 256 мегабайт

Вася называет граф *смешным*, если разность степеней любых двух вершин не превосходит 1 по абсолютной величине. У Васи уже есть некоторый *смешной* граф, и он хочет удалить несколько ребер из графа таким образом, чтобы получившийся граф также являлся *смешным*, а также степень хотя бы одной вершины полученного графа была равна  $d$ .

Помогите Васе построить такой граф.

**Входные данные**

В первой строке записаны три числа  $N$  ( $1 \leq N \leq 300$ ) — количество вершин в графе,  $M$  ( $1 \leq M \leq \frac{N(N-1)}{2}$ ) — количество ребер в графе и  $d$  ( $0 \leq d \leq 299$ ) — требуемая степень.

В каждой из следующих  $M$  строк содержатся два различных числа — номера вершин, соединенных  $i$ -м ребром. Гарантируется, что в графе нет кратных ребер, а также, что граф является *смешным*.

### Выходные данные

Если у Васи не получится построить желаемый граф, в единственной строке выведите «NO» (без кавычек).

В противном случае в первой строке выведите «YES» (без кавычек).

Во второй строке выведите количество ребер в полученном графе.

В третьей строке выведите номера ребер в полученном графе в **возрастающем** порядке. Ребра нумеруются в порядке ввода, начиная с единицы.

Если существует несколько решений, выведите любое.

входные данные
5 6 1 1 2 2 3 1 3 1 4 4 5 3 5
выходные данные
YES 2 2 5

входные данные
4 3 3 1 2 2 3 3 4
выходные данные
NO

## 1Н. Паросочетание vs независимое множество

1 секунда, 256 мегабайт

Дан неориентированный граф, состоящий из  $3 \cdot n$  вершин и  $m$  ребер. Требуется найти паросочетание, состоящее из  $n$  ребер, **или** независимое множество, состоящее из  $n$  вершин.

Множество ребер называется паросочетанием, если никакие два ребра этого множества не имеют общих вершин.

Множество вершин называется независимым, если никакие две вершины этого множества не соединены ребром.

### Входные данные

В первой строке записано одно целое число  $T \geq 1$  — количество графов, которое вам надо обработать. Затем следуют описания  $T$  графов.

В первой строке описания графа записаны два числа  $n$  и  $m$ , где  $3 \cdot n$  — количество вершин, а  $m$  — количество ребер в графе ( $1 \leq n \leq 10^5$ ,  $0 \leq m \leq 5 \cdot 10^5$ ).

В следующих  $m$  строках записаны пары чисел  $v_i$  и  $u_i$  ( $1 \leq v_i, u_i \leq 3 \cdot n$ ), означающие, что вершины  $v_i$  и  $u_i$  соединены ребром.

Гарантируется, что в графе нет петель и кратных ребер.

Гарантируется, что сумма  $n$  по всем графам в одном тесте не превышает  $10^5$ , сумма  $m$  по всем графам в одном тесте не превышает  $5 \cdot 10^5$ .

### Выходные данные

Выведите ответы для каждого из  $T$  графов. Формат ответа для одного графа описан ниже.

Если вы нашли паросочетание размера  $n$ , то в первой строке выведите «Matching» (без кавычек), а во второй строке выведите  $n$  чисел, разделённых пробелами, — номера рёбер паросочетания. Рёбра нумеруются от 1 до  $m$  в порядке ввода.

Если вы нашли независимое множество размера  $n$ , то в первой строке выведите «IndSet» (без кавычек), а во второй строке выведите  $n$  чисел, разделённых пробелами, — номера вершин независимого множества.

Если в графе нет ни того, ни другого, выведите «Impossible» (без кавычек).

Номера рёбер или вершин можно выводить в произвольном порядке.

Если существует несколько решений, вы можете вывести любое. В том числе, если в графе есть как паросочетание размера  $n$ , так и независимое множество размера  $n$ , то вы можете вывести любое из таких паросочетаний **или** любое из таких независимых множеств.

#### ВХОДНЫЕ ДАННЫЕ

```
4
1 2
1 3
1 2
1 2
1 3
1 2
2 5
1 2
3 1
1 4
5 1
1 6
2 15
1 2
1 3
1 4
1 5
1 6
2 3
2 4
2 5
2 6
3 4
3 5
3 6
4 5
4 6
5 6
```

#### ВЫХОДНЫЕ ДАННЫЕ

```
Matching
2
IndSet
1
IndSet
2 4
Matching
1 15
```

Первые два графа из примера совпадают, однако в этом графе есть как паросочетание размера 1, так и независимое множество размера 1. Любое из таких паросочетаний и независимых множеств является правильным ответом.

В третьем графе нет паросочетания размера 2, однако есть независимое множество размера 2. В этом графе также есть независимое множество размера 5: 2 3 4 5 6. Тем не менее, такой ответ не является верным, потому что вас просят найти независимое множество (или паросочетание) размера **ровно**  $n$ .

В четвёртом графе нет независимого множества размера 2, однако есть паросочетание размера 2.

## 11. Минимальное контролирующее множество

2 секунды, 64 мегабайта

Требуется построить в двудольном графе минимальное контролирующее множество, если дано максимальное паросочетание.

### Входные данные

В первой строке файла даны два числа  $m$  и  $n$  ( $1 \leq m, n \leq 4\,000$ ) — размеры долей. Каждая из следующих  $m$  строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа  $K_i$  ( $0 \leq K_i \leq n$ ) — количества ребер, после которого записаны вершины второй доли, соединенные с данной вершиной первой доли, в произвольном порядке. Сумма всех  $K_i$  во входном файле не превосходит 500 000. Последняя строка файла содержит некоторое максимальное паросочетание в этом графе —  $m$  чисел  $0 \leq L_i \leq n$  — соответствующая  $i$ -й вершине первой доли вершина второй доли, или 0, если  $i$ -я вершина первой доли не входит в паросочетание.

### Выходные данные

Первая строка содержит размер минимального контролирующего множества. Вторая строка содержит количество вершин первой доли  $S$ , после которого записаны  $S$  чисел — номера вершин первой доли, входящих в контролирующее множество, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

### ВХОДНЫЕ ДАННЫЕ

```
3 2
2 1 2
1 2
1 2
1 2 0
```

### ВЫХОДНЫЕ ДАННЫЕ

```
2
1 1
1 2
```

## 2A. Максимальное паросочетание в произвольном графе

2 секунды, 256 мегабайт

Ден неориентированный граф, состоящий из  $n$  вершин и  $m$  ребер. Требуется найти максимальное паросочетание в данном графе.

### Входные данные

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n \leq 500$ ,  $0 \leq m \leq \frac{n(n-1)}{2}$ ) — количество вершин и ребер в графе.

Каждая из следующих  $m$  строк содержит два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ) — номера вершин, соединенных  $i$ -м ребром.

Гарантируется, что граф не содержит петель и кратных ребер.

### Выходные данные

В первой строке выведите одно целое число  $k$  — размер максимального паросочетания в данном графе.

В каждой из следующих  $k$  строк выведите два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ) — номера вершин, соединенных  $i$ -м ребром, вошедшим в паросочетание.

входные данные
7 8 3 1 1 6 6 7 7 2 2 1 2 4 4 5 2 5
выходные данные
3 1 3 2 7 4 5

входные данные
5 4 1 2 1 3 1 4 1 5
выходные данные
1 1 2

3А. Максимальный поток на минималках

1 секунда, 256 мегабайт

Костя поспорил с Ваней, что сможет написать самый примитивный поиск максимального потока за 15 минут. К сожалению, Костя прослушал лекцию о поиске максимального потока, поэтому теперь ему нужна ваша помощь.

**Входные данные**  
В первой строке входного файла содержится два числа:  $n$  и  $m$  ( $2 \leq n \leq 10, 1 \leq m \leq n \cdot (n - 1)$ ). Это количество вершин и рёбер в графе, в котором вам требуется найти поток. Далее следуют описания рёбер графа, по одному в каждой строке входного файла. Описание ребра состоит из трёх чисел:  $a, b, c$  ( $1 \leq a, b \leq n, a \neq b, 1 \leq c \leq 100$ ). Эти числа означают, что из вершины  $a$  в вершину  $b$  идёт ребро пропускной способности  $c$ . Гарантируется, что в графе нет кратных рёбер.

**Выходные данные**  
В единственную строку выходного файла выведите одно число — размер максимального потока из вершины  $1$  в вершину  $n$ .

входные данные
4 5 1 2 2 1 3 3 3 2 1 2 4 3 3 4 2
выходные данные
5

3В. Максимальный поток

0.25 секунд, 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером  $1$  в вершину с номером  $n$ .



### Входные данные

Первая строка входного файла содержит  $n$  и  $m$  — число вершин и ребер в графе ( $2 \leq n \leq 500$ ,  $1 \leq m \leq 10\,000$ ). Последующие строки описывают ребра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят  $10^9$ .

### Выходные данные

Выведите величину максимального потока между вершинами 1 и  $n$ .

входные данные
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1
выходные данные
3

## 3C. Разрез

1 секунда, 64 мегабайта

Дан неориентированный граф. Найдите минимальный разрез между вершинами 1 и  $n$ .

### Входные данные

На первой строке входного файла содержится  $n$  ( $1 \leq n \leq 100$ ) — число вершин в графе и  $m$  ( $0 \leq m \leq 400$ ) — количество ребер. На следующих  $m$  строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее 10 000 000), при этом никакие две вершины не соединяются более чем одним ребром.

### Выходные данные

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

входные данные
6 8 1 2 3 1 3 3 2 4 2 2 5 2 3 4 2 3 5 2 5 6 3 4 6 3
выходные данные
2 6 1 2

## 3D. Улиточки

2 секунды, 256 мегабайт

Две улиточки Маша и Петя сейчас находятся на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до  $n$  и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). Ввиду соображений гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

### Входные данные

В первой строке файла записаны четыре целых числа —  $n$ ,  $m$ ,  $a$  и  $h$  (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика).

В следующих  $m$  строках записаны пары чисел. Пара чисел  $(x, y)$  означает, что есть дорожка с лужайки  $x$  до лужайки  $y$  (из-за особенностей улиток и местности дорожки односторонние).

Ограничения:  $2 \leq n \leq 10^5, 0 \leq m \leq 10^5, a \neq h$

**Выходные данные**

Если существует решение, то выведите YES и на двух отдельных строчках сначала путь для Машеньки (т.к. дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

входные данные
3 3 1 3 1 2 1 3 2 3
выходные данные
YES 1 3 1 2 3

3E. Декомпозиция потока

2 секунды, 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером  $n$  и постройте декомпозицию этого потока.

**Входные данные**

Первая строка входного файла содержит  $n$  и  $m$  — количество вершин и количество ребер графа ( $2 \leq n \leq 500, 1 \leq m \leq 10000$ ). Следующие  $m$  строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят  $10^9$ .

**Выходные данные**

В первую строку выходного файла выведите одно число — количество путей в декомпозиции максимального потока из вершины с номером 1 в вершину с номером  $n$ . Следующий строки должны содержать описания элементарных потоков, на который был разбит максимальный. Описание следует выводить в следующем формате: величина потока, количество ребер в пути, вдоль которого течет данный поток и номера ребер в этом пути. Ребра нумеруются с единицы в порядке появления во входном файле.

входные данные
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1
выходные данные
3 1 2 2 5 1 3 2 3 4 1 2 1 4

3F. Химия!!!

2 секунды, 64 мегабайта

Вася и Сережа играют в следующую игру. В некоторых клетках клетчатого листка Сережа рисует один из символов 'H', 'O', 'N' или 'C', после чего Вася должен провести между некоторыми находящимися в соседних клетках символами линии так, чтобы получилось корректное изображение химической молекулы. К сожалению, Сережа любит рисовать много символов, и Вася не может сразу определить, возможно ли вообще нарисовать линии нужным способом. Помогите ему написать программу, которая даст ответ на этот вопрос.

В этой задаче проведенные между символами химических элементов линии будем считать корректным изображением молекулы, если они удовлетворяют следующим условиям:

- каждая линия соединяет символы, нарисованные в соседних (по стороне) клетках,
- между каждой парой символов проведено не более одной линии,
- от каждого элемента отходит ровно столько линий, какова валентность этого элемента (1 для H, 2 для O, 3 для N, 4 для C),
- пустые клетки ни с чем не соединены, и
- хотя бы в одной клетке нарисован какой-то символ.

**Входные данные**

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  ( $1 \leq n, m \leq 50$ ) — размеры листочка, на котором рисует Сережа. Далее следуют  $n$  строк по  $m$  символов в каждой, задающих конфигурацию химических элементов, которую нарисовал Сережа; пустые клетки задаются символом '.'.

**Выходные данные**

В выходной файл выведите одно слово: 'Valid', если линии провести требуемым образом можно, и 'Invalid', если нельзя.

входные данные
3 4 НОН. NCOH OO..
выходные данные
Valid

входные данные
3 4 НОН. NCOH OONH

выходные данные
Invalid

**3G. Живопись**

2 секунды, 256 мегабайт

В стране Олимпия очень развита живопись. Картиной считается любой прямоугольник, который состоит из черных и белых единичных квадратов. Художник Олимпус решил радикально улучшить свои картины. Для этого он планирует к белому и черному цветам добавить еще и серый оттенок. По его задумке, граница между каждым черным и белым квадратом должна содержать серую линию, чтобы образовался эффект плавного перехода.

Однако, перед началом работы, он обнаружил, что серая краска очень дорого стоит. Чтобы сэкономить деньги художник решил оценить, не выгоднее ли сначала перекрасить некоторые белые квадраты в черные, а черные в белые для того, чтобы минимизировать расходы на краску.

Напишите программу, которая по информации о существующей картине определяет минимальную сумму денег, которые понадобятся на ее улучшение.

**Входные данные**

Первая строка входного файла содержит пять натуральных чисел  $N$ ,  $M$ ,  $w$ ,  $b$ ,  $g$ . ( $1 \leq N, M \leq 70$ ) — высота и ширина картины, ( $1 \leq w, b, g \leq 1000$ ) — цена рисования одного белого единичного квадрата, черного единичного квадрата и серой линии единичной длины, соответственно.

Далее следует  $N$  строк, каждая из которых состоит из  $M$  литер. Литера B соответствует черному квадрату, а W — белому.

**Выходные данные**

Единственная строка выходного файла должна содержать одно целое число, которое есть минимальной суммой затрат на улучшение картины.

входные данные
3 2 10 12 1 BW WB BW
выходные данные
7

### 3Н. Чаепитие

2 секунды, 256 мегабайт

В одном из отделов крупной организации работает  $n$  человек. Как практически все сотрудники этой организации, они любят пить чай в перерывах между работой. При этом они достаточно дисциплинированы и делают в день ровно один перерыв, во время которого пьют чай. Для того, чтобы этот перерыв был максимально приятным, каждый из сотрудников этого отдела обязательно пьет чай одного из своих любимых сортов. В разные дни сотрудник может пить чай разных сортов. Для удобства пронумеруем сорта чая числами от 1 до  $m$ .

Недавно сотрудники отдела купили себе большой набор чайных пакетиков, который содержит  $a_1$  пакетиков чая сорта номер 1,  $a_2$  пакетиков чая сорта номер 2, ...,  $a_m$  пакетиков чая сорта номер  $m$ . Теперь они хотят знать, на какое максимальное число дней им может хватить купленного набора так, чтобы в каждый из дней каждому из сотрудников доставался пакетик чая одного из его любимых сортов.

Каждый сотрудник отдела пьет в день ровно одну чашку чая, которую заваривает из одного пакетика. При этом пакетики чая не завариваются повторно.

### Входные данные

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 50$ ). Вторая строка содержит  $m$  целых чисел  $a_1, \dots, a_m$  ( $1 \leq a_i \leq 10^6$  для всех  $i$  от 1 до  $m$ ).

Далее следуют  $n$  строк —  $i$ -я из этих строк описывает любимые сорта  $i$ -го сотрудника отдела и имеет следующий формат: сначала следует положительное число  $k_i$  — количество любимых сортов чая этого сотрудника, а затем идут  $k_i$  различных чисел от 1 до  $m$  — номера этих сортов.

### Выходные данные

Выведите одно целое число — искомое максимальное количество дней.

входные данные
2 3 3 2 1 2 1 2 2 1 3
выходные данные
3

### 3I. Чокнутый профессор

2 секунды, 256 мегабайт

В Университете города М. проводят эксперимент. Преподаватели сами решают, что они будут читать в рамках того или иного курса. И вот преподаватель математического анализа (в простонародье — матана) оценил по некоторым критериям все известные ему темы в данном курсе. В результате этой ревизии каждой теме сопоставлено некоторое целое число (возможно, отрицательное) — полезность данной темы. Профессор хочет максимизировать суммарную полезность прочитанных им тем, но не все так просто. Для того чтобы студенты поняли некоторые темы, необходимо, чтобы были прочитаны так же некоторые другие темы, так как некоторые доказательства базируются на фактах из других тем. Однако если существует цикл из зависимостей тем, то их все можно прочитать, и на качестве понимания материала студентами это не скажется.

Вас попросили составить список тем, которые профессор должен прочитать, таким образом, чтобы студенты все поняли, и суммарная полезность курса была максимальна.

### Входные данные

Первая строка входного файла содержит одно число —  $N$  ( $1 \leq N \leq 200$ ). Вторая строка содержит  $N$  целых чисел, не превосходящих по модулю 1 000 — полезности каждой темы. Далее следуют  $N$  строк с описанием зависимостей тем. Каждое описание начинается количеством тем, которые необходимо понять для понимания данной темы. Потом следуют номера этих тем, разделенные пробелами.

### Выходные данные

Выведите единственное число — максимально возможную суммарную полезность прочитанного материала.

входные данные
4 -1 1 -2 2 0 1 1 2 4 2 1 1

выходные данные
2

входные данные
3 2 -1 -2 2 2 3 0 0
выходные данные
0

## 4А. Максимальный поток минимальной стоимости

2 секунды, 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает пропускной способностью и стоимостью. Найдите максимальный поток минимальной стоимости из вершины с номером 1 в вершину с номером  $n$ .

### Входные данные

Первая строка входного файла содержит  $n$  и  $m$  — количество вершин и количество ребер графа ( $2 \leq n \leq 100$ ,  $0 \leq m \leq 1000$ ). Следующие  $m$  строк содержат по четыре целых числа: номера вершин, которые соединяет соответствующее ребро графа, его пропускную способность и его стоимость. Пропускные способности и стоимости неотрицательны и не превосходят  $10^5$ .

### Выходные данные

В выходной файл выведите одно число — цену максимального потока минимальной стоимости из вершины с номером 1 в вершину с номером  $n$ . Ответ не превышает  $2^{63} - 1$ .

входные данные
4 5 1 2 1 2 1 3 2 2 3 2 1 1 2 4 2 1 3 4 2 3
выходные данные
12

## 4B. Задача о назначениях

1.5 секунд, 256 мегабайт

Дана целочисленная матрица  $C$  размера  $n \times n$ . Требуется выбрать  $n$  ячеек так, чтобы в каждой строке и каждом столбце была выбрана ровно одна ячейка, а сумма значений в выбранных ячейках была минимальна.

### Входные данные

Первая строка входного файла содержит  $n$  ( $2 \leq n \leq 300$ ). Каждая из последующих  $n$  строк содержит по  $n$  чисел:  $C_{ij}$ . Все значения во входном файле неотрицательны и не превосходят  $10^6$ .

### Выходные данные

В первую строку выходного файла выведите одно число — искомая минимизируемая величина. Далее выведите  $n$  строк по два числа в каждой — номер строки и столбца клетки, участвующей в оптимальном назначении.

Пары чисел можно выводить в произвольном порядке.

входные данные
3 3 2 1 1 3 2 2 1 3

выходные данные
3 2 1 3 2 1 3

Обратите внимание, что эту задачу можно сдать с помощью алгоритма за  $O(n^3)$ , без использования Венгерского алгоритма, а используя только максимальный поток минимальной стоимости.

## 4C. Командная олимпиада

1 секунда, 256 мегабайт

Совсем скоро начнется первый тур очередной всероссийской командной олимпиады школьников по палеонтологии (ВКОШП). На олимпиаду приехали команды из  $n$  школ, от каждой школы приехало ровно по две команды. Команды уже заняли свои места, когда обнаружилось, что некоторые команды из одной школы сидят слишком близко. Перед организаторами олимпиады встала серьезная задача — пересадить участников олимпиады.

Столбы, за которыми сидят команды, расставлены в один ряд. Расстояние между рабочими местами соседних команд оказалось равно 10 метрам. Организаторы хотят, чтобы минимальное расстояние между двумя командами из одной школы было как можно больше.

Пересаживая команду, организатором необходимо перенести на новое место все оборудование, приготовленное командой для исследований. Поэтому организаторы хотят пересадить команды так, чтобы сумма расстояний между старыми и новыми рабочими местами команд была как можно меньше.

Например, пусть в соревновании принимают участие по две команды школ 1, 2, 3 и 4. Пусть исходно команды распределены по рабочим местам следующим образом: 1, 3, 2, 2, 1, 4, 4, 3 (для каждой команды указан номер школы, которую она представляет). При таком распределении по рабочим местам команды из школы 2 сидят слишком близко, как и команды из школы 4. Пересадив команды в следующем порядке: 1, 3, 2, 4, 1, 3, 2, 4, жюри может добиться своего: команды из одной школы сидят на местах, расстояние между которыми не меньше 40 м, большего расстояния добиться нельзя. Сумма расстояний между старыми и новыми позициями для данного примера равна  $0 + 0 + 0 + 20 + 0 + 20 + 30 + 10 = 80$  м, для исходного распределения команд она минимальна.

Вам задано исходное распределение команд по рабочим местам. Требуется пересадить их так, чтобы минимальное расстояние между командами из одной школы было как можно больше. При этом среди возможных новых размещений команд следует выбрать такое, чтобы сумма расстояний между старыми и новыми местами рабочими команд была минимально возможной.

### Входные данные

В первой строке входного файла задано число  $n$  — количество команд ( $1 \leq n \leq 100$ ). Во второй строке задано исходное распределение команд по рабочим местам — последовательность  $a_1, a_2, \dots, a_{2n}$ . Для каждой команды указан номер школы, которую она представляет. Гарантируется, что последовательность состоит из чисел от одного до  $n$  и каждое число встречается ровно два раза.

### Выходные данные

В единственную строку выходного файла выведите, каким образом следует пересадить команды, чтобы минимальное расстояние между командами из одной школы было как можно больше. При этом среди возможных новых размещений команд следует выбрать такое, чтобы сумма расстояний между старыми и новыми рабочими местами команд была минимально возможной. Если оптимальных ответов несколько, можно вывести любой из них.

входные данные
4 1 3 2 2 1 4 4 3
выходные данные
1 2 4 3 1 2 4 3

## 4D. Расшифровка

1 секунда, 256 мегабайт

Недавно на уроке во время контрольной Мария Ивановна перехватила записку от Саши к Оле. Мария Ивановна очень хочет знать, что в записке, но, к сожалению, записка зашифрована. Мария Ивановна знает, что её ученики для шифровки заменяют каждую букву исходного сообщения на какую-то другую. Замена происходит таким образом, что одинаковые буквы всегда заменяются одной и той же буквой, а разные — разными.

Мария Ивановна подозревает, что записка — это ответы к контрольному тесту (ведь её длина случайно оказалась равной длине строки с правильными ответами). Однако она знает, что ответы Саши не обязательно полностью правильны. На каждый вопрос возможен один из  $K$  вариантов ответа. Естественно, Мария Ивановна знает правильные ответы.

Мария Ивановна решила расшифровать записку таким способом, чтобы максимизировать количество правильных ответов Саши. Однако, она очень занята, поэтому попросила Вас помочь ей в этом пустяковом деле.

### Входные данные

В первой строке задана длина каждой из строк  $N$  ( $1 \leq N \leq 2\,000\,000$ ) и  $K$  — количество возможных ответов на каждый вопрос ( $1 \leq K \leq 52$ ). Ответы нумеруются в порядке  $abcde \dots xyzABCDE \dots XYZ$ . То есть, при  $K = 6$  возможные ответы выглядят как  $abcdef$ , а при  $K = 30$  —  $abcde \dots xyzABCD$ .



Во второй строке задана зашифрованная записка — строка, состоящая из строчных и заглавных латинских букв.

В третьей строке заданы правильные ответы — строка той же длины, что и первая, состоящая из строчных и заглавных латинских букв.

### Выходные данные

В первой строке выведите единственное число — максимально возможное количество правильных ответов у Саши.

Во второй строке выведите расшифровку — строчку длины  $K$ , где по порядку для каждой буквы из шифра учеников указано, какому ответу она соответствует.

Если несколько расшифровок дают правильный ответ, выведите любую.

входные данные
10 2 aaabbbbaaab bbbbabbbbbb
выходные данные
7 ba

входные данные
10 2 aaaaaaabbb bbbbaaabbb
выходные данные
6 ab

входные данные
9 4 dacbdacbd acbdacbdba

### выходные данные

9  
cdba

## 4Е. Автоматное программирование

5 секунд, 256 мегабайт

В один замечательный день в компанию «Х» завезли  $k$  автоматов. И не простых автоматов, а автоматов-программистов! Это был последний неудачный шаг перед переходом на андроидов-программистов, но это уже совсем другая история.

В компании сейчас  $n$  задач, для каждой из которых известно время начала ее выполнения  $s_i$ , длительность ее выполнения  $t_i$  и прибыль компании от ее завершения  $c_i$ . Любой автомат может выполнять любую задачу, ровно одну в один момент времени. Если автомат начал выполнять задачу, то он занят все моменты времени с  $s_i$  по  $s_i + t_i - 1$  включительно и не может переключиться на другую задачу.

Вам требуется выбрать набор задач, которые можно выполнить с помощью этих  $k$  автоматов и который принесет максимальную суммарную прибыль.

### Входные данные

В первой строке записаны два целых числа  $n$  и  $k$  ( $1 \leq n \leq 1000$ ,  $1 \leq k \leq 50$ ) — количество задач и количество автоматов, соответственно.

В следующих  $n$  строках через пробелы записаны тройки целых чисел  $s_i, t_i, c_i$  ( $1 \leq s_i, t_i \leq 10^9$ ,  $1 \leq c_i \leq 10^6$ ),  $s_i$  — время начала выполнения  $i$ -го задания,  $t_i$  — длительность  $i$ -го задания, а  $c_i$  — прибыль от его выполнения.

### Выходные данные

Выведите  $n$  целых чисел  $x_1, x_2, \dots, x_n$ . Число  $x_i$  должно быть равно 1, если задачу  $i$  следует выполнить, и 0 в противном случае.



Если оптимальных решений несколько, то выведите любое из них.

входные данные
3 1 2 7 5 1 3 3 4 1 3
выходные данные
0 1 1

входные данные
5 2 1 5 4 1 4 5 1 3 2 4 1 2 5 6 1
выходные данные
1 1 0 0 1

Условие  
недоступно  
на  
русском  
языке

## 5B. Точка в углу

1 секунда, 256 мегабайт

Дан угол  $AOB$  ( $O$  - вершина угла,  $A$  и  $B$  - точки на сторонах) и точка  $P$ . Определите, принадлежит ли точка  $P$  углу  $AOB$  (включая его стороны: лучи  $OA$  и  $OB$ ).

## Входные данные

Программа получает на вход координаты точек  $A, O, B, P$ . Все координаты - целые, не превосходят  $10^5$  по модулю. Точки  $A, O, B$  не лежат на одной прямой.

## Выходные данные

Программа должна вывести слово *YES* или *NO*.

входные данные
0 1 0 0 1 0 1 1
выходные данные
YES

входные данные
1 0 0 0 0 1 -1 -1
выходные данные
NO

входные данные
-1 5 3 3 0 0 -1 1
выходные данные
YES

## 5C. Пересечение отрезков

1 секунда, 256 мегабайт

Даны два отрезка:  $AB$  и  $CD$ . Определите, какое множество точек является пересечением этих отрезков.

**Входные данные**

Программа получает на вход восемь целых чисел, по абсолютной величине не превосходящих  $10^4$  — координаты точек  $A, B, C, D$ . Точки могут совпадать (в том числе могут совпадать и концы одного отрезка).

**Выходные данные**

Если указанные отрезки не пересекаются, то выведите строку «Empty». Если отрезки пересекаются в одной точке, то выведите два числа — координаты точки пересечения. Если пересечением является отрезок, то выведите четыре числа — координаты двух концов отрезка в лексикографическом порядке (то есть сначала нужно вывести ту точку, у которой меньше координата  $x$ , а если у них равны координаты  $x$ , то ту, у которой меньше координата  $y$ ). Все числа следует выводить с точностью не менее 6 знаков после запятой.

входные данные
0 0 9 9 9 5 0 5
выходные данные
5.0000000000 5.0000000000

входные данные
0 0 9 9 15 15 7 7
выходные данные
7.0000000000 7.0000000000 9.0000000000 9.0000000000

входные данные
0 0 9 9 10 10 10 10
выходные данные
Empty

5D. Прямая и окружность

1 секунда, 256 мегабайт

**Входные данные**

Шесть чисел — координаты центра и радиус окружности и коэффициенты  $A, B$  и  $C$  нормального уравнения прямой. Все числа целые и не превосходят  $10^4$ .

**Выходные данные**

В первой строке одно число  $K$ , равное количеству точек пересечения прямой с окружностью. Далее в  $K$  строках координаты самих точек в любом порядке.

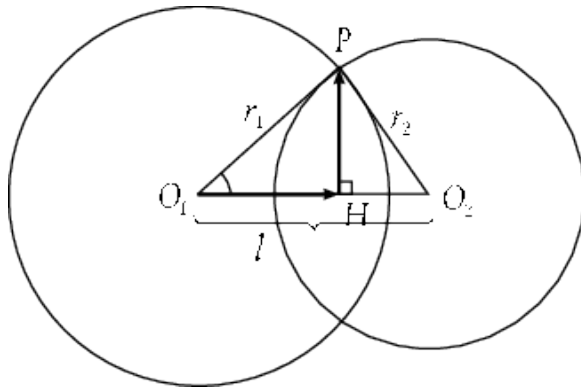
входные данные
2 3 1 1 -1 0
выходные данные
2 2.00000000 2.00000000 3.00000000 3.00000000

входные данные
2 4 1 1 -1 0
выходные данные
0

## 5Е. Пересекаем окружности

1 секунда, 256 мегабайт

Вам заданы две окружности. Необходимо выяснить, пересекаются ли заданные окружности и найти точки их пересечения.



### Входные данные

Первая строка файлов содержит количество тестов — целое число от 1 до 10 000.

Далее следуют тесты. Каждый тест имеет вид:

Две строки, по три числа в каждой — координаты центра и радиус сначала первой, а затем второй окружности.

Все числа — целые, по модулю не превосходящие 10 000. А радиус еще и положительный.

### Выходные данные

На каждый тест выведите ответ в следующем формате:

На отдельной строке выходного файла выведите количество точек пересечения (0, 1, 2 или 3, если их бесконечно много). В случае одной точки пересечения выведите во второй строке координаты этой точки. В случае двух точек пересечения выведите во второй строке

координаты точки  $H$ , в третьей длины векторов  $\overrightarrow{O_1H}$  и  $\overrightarrow{HP}$ , в следующих двух строках должны находиться координаты точек пересечения. Эти две точки можно вывести в произвольном порядке.

### входные данные

```
4
3 4 5
11 4 2
3 4 5
11 4 3
3 4 5
11 4 4
3 4 5
3 4 5
```

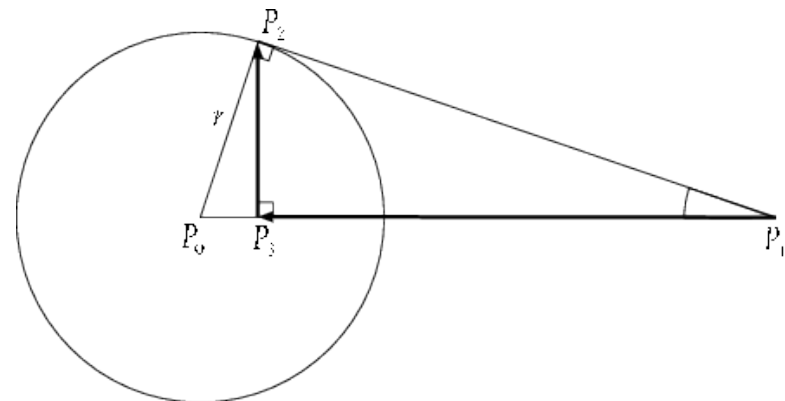
### выходные данные

```
0
1
8.0000000000 4.0000000000
2
7.5625000000 4.0000000000
4.5625000000 2.0453835215
7.5625000000 6.0453835215
7.5625000000 1.9546164785
3
```

## 5F. Касательная к окружности

2 секунды, 256 мегабайт

Вам задана окружность и точка. Точка может лежать вне окружности, на ее границе, а также внутри окружности. Необходимо провести касательные к окружности (если это возможно) и найти точки касания.



## Входные данные

В первой строке входного файла находятся два числа — координаты центра окружности  $P_0$ . Во второй строке записан радиус окружности  $r$  ( $1 \leq r \leq 10^4$ ). В третьей строке находятся два числа — координаты точки  $P_1$ . Координаты целые числа не превышающие по модулю  $10^4$ .

## Выходные данные

В первой строке выходного файла выведите количество точек касания (0, 1 или 2). В случае одной точки касания выведете во второй строке координаты этой точки. В случае двух точек касания выведите во второй строке координаты точки  $P_3$ , в третьей строке длины векторов  $\overrightarrow{P_1 P_3}$  и  $\overrightarrow{P_3 P_2}$ , в следующих двух строках должны находиться координаты точек касания.

Входные данные	
2 2	
2	
2 5	
Выходные данные	
2	
2.0000000000 3.3333333333	
1.6666666667 1.4907119850	
0.5092880150 3.3333333333	
3.4907119850 3.3333333333	

Входные данные	
10 10	
10	
20 20	
Выходные данные	
2	
15.0000000000 15.0000000000	
7.0710678119 7.0710678119	
10.0000000000 20.0000000000	
20.0000000000 10.0000000000	

## 5G. Ежик и грибы

1 секунда, 256 мегабайт

Ежик путешествует по координатной плоскости. Изначально он находится в точке с координатами  $(0, 0)$  и смотрит в направлении оси абсцисс (оси  $Ox$ ).

Цель ежика — собрать ровно  $n$  грибов, причем строго в описанном порядке, то есть сначала первый, затем второй, ..., и только в конце  $n$ -й. Известно, что  $i$ -й гриб растет в точке с координатами  $(x_i, y_i)$ .

Чтобы собрать  $i$ -й гриб, ежик перемещается из своего текущего положения в точку  $(x_i, y_i)$  **строго по прямой**, все время смотря вперед по направлению движения. Определить точное направление движения ежик справится и сам с помощью нюха, но для этого ему сначала придется повернуться хотя бы приблизительно в нужную сторону. Помогите ему в этом!

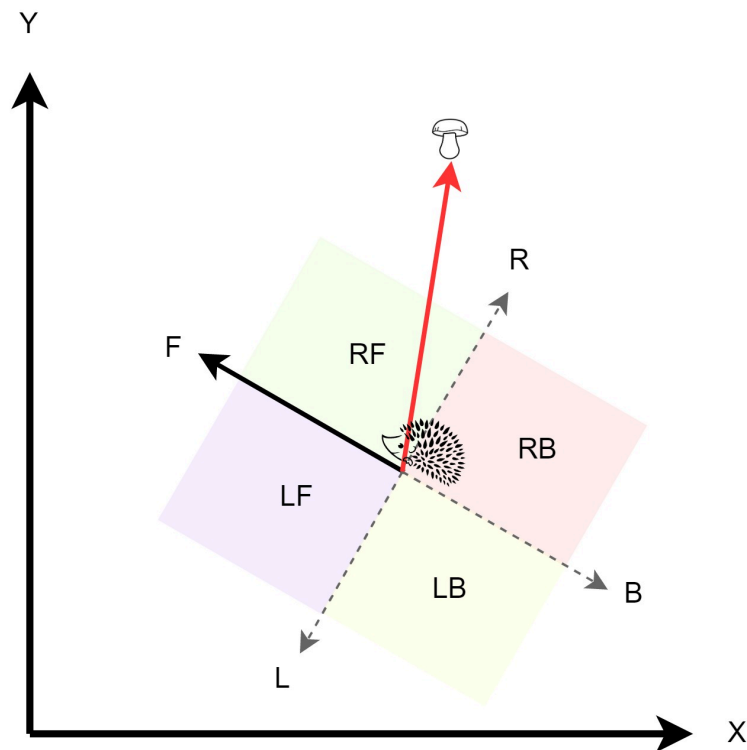
Более формально, для каждого из  $n$  перемещений ежика выведите одну из строк:

- «L» или «R», если для перемещения к следующему грибу ежику следует повернуться ровно на  $90^\circ$  против или по часовой стрелке соответственно;
- «F» или «B», если для перемещения к следующему грибу не требуется поворачиваться вообще или требуется развернуться на  $180^\circ$  соответственно;
- «LF», «LB», «RB» или «RF», если следующий гриб находится строго внутри первой (слева-спереди), второй (слева-сзади), третьей (справа-сзади) или четвертой (справа-спереди) четверти плоскости относительно текущего направления ежика.

Ниже проиллюстрировано, как изменяется направление ежика.

Предположим, он только что съел гриб и смотрит в направлении «F».

Красным цветом указано направление до следующего гриба, поэтому ежу нужно далее следовать в направлении четвертой четверти «RF».



### Входные данные

В первой строке дано единственное целое число  $n$  — количество грибов, которые ежик хочет собрать ( $1 \leq n \leq 10^5$ ).

Во  $i$ -й из следующих  $n$  строк через пробел даны два целых числа  $x_i$  и  $y_i$  — координаты  $i$ -го гриба ( $|x_i|, |y_i| \leq 10^9$ ). Гарантируется, что никакие два гриба не растут в одной точке, и что никакой гриб не растет в точке  $(0, 0)$ .

### Выходные данные

Выведите информацию об  $n$  поворотах, совершаемых ежиком, в формате, описанном в условии. Информацию о последовательных поворотах разделяйте пробелом или переводом строки.

### Входные данные

```
3
1 1
0 2
-3 -1
```

### Выходные данные

```
LF L L
```

### Входные данные

```
7
11 0
7 3
3 3
-6 -4
10 100
-1 -10
2 20
```

### Выходные данные

```
F LB LF LF RB RB B
```

## 5Н. Две самые далёкие точки (наивная версия)

0.25 секунд, 256 мегабайт

### Входные данные

Первая строка содержит количество точек  $N$ , ( $\{1 \leq N \leq 7000\}$ ). Каждая из последующих  $N$  строк содержит два целых числа — координаты  $x_i$  и  $y_i$ . Координаты по модулю не превосходят  $10^9$ .

### Выходные данные

Выведите в выходной файл расстояние между двумя наиболее удалёнными точками с максимально возможной точностью.

входные данные
7 0 0 1 1 2 2 0 2 1 3 0 1 2 0
выходные данные
3.16227766016837952279

входные данные
8 0 0 38 85 40 90 20 96 0 100 -20 96 -40 90 -38 85
выходные данные
100.0000000000000000000000

Не используйте хитрые алгоритмы, напишите наивное решение за  $O(n^2)$

## 51. Битва за Хогвартс

2 секунды, 512 мегабайт

Приспешники Тёмного Лорда уже подошли совсем близко к Хогвартсу.

Хогвартс окружен системой защитных башен. Профессор Флитвик накладывает защитные заклятия на замок. Закljučаются они в следующем: вокруг выпуклой оболочки башен создается защитный барьер.

Тёмный Лорд, пользуясь Бузинной палочкой, может разрушить защитный барьер за минуту, при этом все башни на выпуклой оболочке тоже разрушаются.

После того, как башни уничтожены, Флитвик мгновенно восстанавливает защитный барьер на выпуклой оболочке оставшихся башен, а Волан-де-Морт их снова разрушает через минуту. Так продолжается, пока все башни не падут.

У Гарри и его друзей мало времени — они ищут и уничтожают очередной крестраж. Поэтому их очень интересует, сколько времени у них осталось.

Рассчитайте для каждой башни момент времени, когда она будет уничтожена.

**Входные данные**  
В первой строке вводится одно число  $n$  ( $1 \leq n \leq 2 \cdot 10^4$ ) — количество башен.

В следующих  $n$  строках вводятся целочисленные координаты башен —  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^4$ ).

Гарантируется, что башни расположены так, что каждый следующий защитный барьер будет лежать строго внутри предыдущего (то есть, они не пересекаются и не имеют точек касания).

**Выходные данные**  
Выведите  $n$  строк, каждая из которой содержит одно число — момент времени (в минутах), к которому падёт каждая башня, начиная с первой.

входные данные
5 0 0 4 4 0 4 1 1 4 0

выходные данные

1  
1  
1  
2  
1

5J. Замок

1 секунда, 256 мегабайт

Есть замок — точка  $(0, 0)$ . Замок окружен несколькими непересекающимися заборами, каждый представляет из себя выпуклый многоугольник.

Есть также  $m$  захватчиков, известны их координаты. Захватчики не умеют перелезать через заборы. Захватчика будем считать опасным, если он находится внутри внешнего забора. Требуется вычислить суммарную площадь области, куда опасные захватчики могут добраться без пересечения заборов.

Входные данные

В первой строке задано число  $n$  — количество заборов ( $1 \leq n \leq 100\,000$ ). Далее следуют описания  $n$  заборов. Каждое описание начинается с числа  $k$ , далее следуют  $k$  строк, содержащих по два целых числа  $x$  и  $y$  — координаты вершин ( $|x|, |y| \leq 2 \cdot 10^6$ ). Вершины каждого многоугольника перечисляются в порядке обхода против часовой стрелки. Гарантируется, что точка  $(0, 0)$  лежит внутри каждого забора.

Далее следует число  $m$  ( $0 \leq m \leq 100\,000$ ) — количество захватчиков. В следующих  $m$  строках заданы координаты захватчиков.

Суммарное число вершин во всех многоугольниках не превосходит 100 000.

Выходные данные

Выведите единственное число — общую захваченную площадь с шестью знаками после десятичной точки.

входные данные

3  
4  
-10 -10  
10 -10  
10 10  
-10 10  
4  
20 20  
-20 20  
-20 -20  
20 -20  
4  
30 -30  
30 30  
-30 30  
-30 -30  
3  
1 1  
22 23  
111 123

выходные данные

2400.000000

5K. Выпуклая оболочка

2 секунды, 256 мегабайт

Дано  $N$  точек на плоскости.

Нужно построить их выпуклую оболочку.

Гарантируется, что выпуклая оболочка не вырождена.

Входные данные

На первой строке число  $N$  ( $3 \leq N \leq 10^5$ ). Следующие  $N$  строк содержат пары целых чисел  $x$  и  $y$  ( $-10^9 \leq x, y \leq 10^9$ ) — точки.

Будьте аккуратны! Точки произвольны. Бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

### Выходные данные

В первой строке выведите  $N$  число вершин выпуклой оболочки. Следующие  $N$  строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой. Кроме того, в последней строке выведите площадь получившейся выпуклой оболочки. Площадь необходимо вывести абсолютно точно.

входные данные
5 0 0 2 0 0 2 1 1 2 2
выходные данные
4 0 0 0 2 2 2 2 0 4.0

входные данные
3 0 0 0 1 1 1
выходные данные
3 0 0 0 1 1 1 0.5

## 5L. Стена

1 секунда, 64 мегабайта

{64 мегабайта}

Жил-был жадный король, который однажды приказал главному архитектору окружить королевский замок стеной. Король был настолько жаден, что не желал слушать рассказы архитектора о красивой кирпичной стене с прекрасным силуэтом и изящными высокими башнями. Вместо этого он приказал окружить замок стеной, затратив минимальное количество камня и времени, но потребовал, чтобы стена не подходила к замку ближе, чем на заданное расстояние. Если король узнает, что архитектор потратил не минимально возможное количество ресурсов, то архитектор лишится головы. Более того, король потребовал, чтобы архитектор сразу же предложил проект стены с указанием минимального количества ресурсов, необходимых для постройки.

Вы должны помочь архитектору сохранить голову, написав программу для поиска минимальной длины стены, удовлетворяющей условиям короля.

Задачу упрощает то, что замок короля имеет форму многоугольника и расположен на равнине. Архитектор уже ввел систему координат и точно измерил координаты вершин замка в футах.

### Входные данные

Первая строка входного файла содержит числа  $N$  и  $L$ , разделенные пробелом.  $N$  ( $3 \leq N \leq 1000$ ) — это количество вершин в королевском замке, а  $L$  ( $1 \leq L \leq 1000$ ) — минимальное количество футов, на которое стена может приближаться к замку.

Следующие  $N$  строк описывают координаты замка в порядке обхода по часовой стрелке. В каждой строке через пробел записаны целые числа  $x_i$  и  $y_i$ , разделенные пробелом ( $-10000 \leq x_i, y_i \leq 10000$ ), которые обозначают координаты  $i$ -ой вершины. Все вершины различны, и никакие две стороны не пересекаются кроме как по вершинам.



## Выходные данные

Выведите минимальную длину стены в футах, удовлетворяющей условиям короля с точностью не менее 6 знаков после запятой.

входные данные
9 100 200 400 300 400 300 300 400 300 400 400 500 400 500 200 350 200 200 200
выходные данные
1628.3185307180

## 5M. Львы (но не тигры!)

1 секунда, 256 мегабайт

Иван — профессиональный фотограф. В погоне за хорошим кадром он готов ехать в любую точку мира. И сейчас он приехал в Уганду с целью сфотографировать львов в их естественной среде обитания. К сожалению, львов не очень заинтересовали профессиональные качества Ивана, зато их весьма заинтересовали его вкусовые качества.

Для простоты будем считать саванну бесконечной плоскостью, а Ивана и львов — точками. В начальный момент времени Иван выбирает направление, в котором он будет ехать, спасаясь от львов, и дальше все время едет по прямой. Львы, будучи великолепными охотниками, сразу определяют направление, выбранное Иваном, и планируют его поимку соответственно. В частности, львы понимают, что Иван поедет по прямой. Скорости всех львов постоянны и равны между собой. Как мы уже увидели, Иван не очень умный, поэтому фотографировать львов он отправился в медленном автомобиле, скорость которого также постоянна и равна скорости львов.

Помогите Ивану выбрать такое направление, чтобы его не догнал ни один лев.

### Входные данные

В первой строке входных данных дано целое число  $n$  — количество львов ( $1 \leq n \leq 5 \cdot 10^5$ ).

Во второй строке даны два целых числа — координаты Ивана.

Далее следуют  $n$  строк, в каждой из которых даны два целых числа — координаты львов.

Все координаты не превосходят  $10^8$  по абсолютной величине. Гарантируется, что все точки во входных данных различны.

### Выходные данные

Если Иван не может убежать, выведите одно слово «NO» (без кавычек). Иначе в первой строке выведите одно слово «YES» (тоже без кавычек).

Составителю контеста было лень написать чекер, поэтому восстанавливать направление не требуется.

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Оценка
0	0	Тесты из условия	подзадача
1	37	$1 \leq n \leq 5000$	подзадача
2	63	—	подзадача

<b>входные данные</b>
2 0 0 1 1 2 2
<b>выходные данные</b>
YES

<b>входные данные</b>
4 0 0 1 1 -1 -1 1 -1 -1 1
<b>выходные данные</b>
NO

### 6A. Теодор Рузвельт

2 секунды, 64 мегабайта

«Теодор Рузвельт» — флагман военно-морского флота Кукуляндии. Заклятые враги кукуляндцев, флатландцы, решили уничтожить его. Они узнали, что «Теодор Рузвельт» представляет собой выпуклый многоугольник из  $n$  вершин и узнали его координаты. Затем они выпустили  $m$  баллистических ракет и определили координаты точек, где эти ракеты взорвались. По расчётам штаба флатландцев, «Теодор Рузвельт» будет уничтожен, если в него попадёт хотя бы  $k$  ракет. Вычислите, удалось ли флатландцам уничтожить корабль.

### Входные данные

В первой строке через пробел записаны целые числа  $n, m, k$  ( $3 \leq n \leq 10^5, 0 \leq k \leq m \leq 10^5$ ). В последующих  $n$  строках записаны координаты вершин многоугольника в порядке обхода против часовой стрелки. В следующих  $m$  строках записаны координаты точек. Гарантируется, что все координаты — целые числа, не превосходящие по модулю  $10^9$ .

### Выходные данные

Выведите «YES», если в многоугольнике или на его границе лежит по крайней мере  $k$  точек, и «NO» в противном случае.

<b>входные данные</b>
5 4 2 1 -1 1 2 0 4 -1 2 -1 -1 -2 -1 1 -1 0 1 2 3
<b>выходные данные</b>
YES

### 6B. Разрезание торта

1 секунда, 64 мегабайта

Мама купила Пете на день рождения торт в виде выпуклого многоугольника. Торт большой и вкусный, и Петя хочет разделить его с мамой поровну. Для этого он хочет сделать один прямолинейный разрез, причем ему будет удобнее, если этот разрез будет параллелен оси  $Oy$ . Помогите Пете определить, как ему разрезать торт.

Входные данные

В первой строке записано целое число  $N$  ( $1 \leq N \leq 10000$ ) — количество вершин многоугольника. В последующих  $N$  строках записаны координаты вершин многоугольника в порядке обхода. Гарантируется, что все координаты — целые числа, не превосходящие по модулю  $10^3$ .

Выходные данные

Выведите  $x$ -координату точки, через которую необходимо провести разрез, с точностью не менее  $10^{-6}$ .

входные данные
4 0 0 0 2 2 2 2 0
выходные данные
1.0000000000

Условие  
недоступно  
на  
русском  
языке

Условие  
недоступно  
на  
русском  
языке

6Е. Апокалипсис

3 секунды, 256 мегабайт

На планете Мидав очень близок конец света. Как известно, эта плоская планета, которую можно представить как бесконечную плоскость с декартовыми координатами. На этой планете есть  $Q$  поселений.

В нулевой день на Мидаве случилось заражение. Оно представляет из себя выпуклый многоугольник на  $N$  вершинах. Каждый день площадь заражения меняется неизвестным образом, но для каждого дня с номером  $i > 0$  верно следующее:

- 1. Если в  $i$ -й день заражена любая точка на расстояние  $d$  от исходного многоугольника, то заражены и все остальные точки на расстоянии не большем  $d$  от исходного многоугольника;
- 2. Пусть  $S_k$  — площадь заражения в  $k$ -й день. Тогда верно  $S_i = 2 \cdot S_{i-1}$ .

Если какое-то поселение окажется внутри или на границе заражения, то все живые организмы в нём сразу же вымрут. Для каждого поселения планеты Мидав осталось совсем немного времени, поэтому ответьте, какой день (включая и нулевой) окажется для поселения последним.

Входные данные

В первой строке дано целое число  $N$  — количество точек в многоугольнике заражения нулевого дня ( $3 \leq N \leq 10^5$ ).

В следующих  $N$  строках даны по два целых числа  $c_{xi}$  и  $c_{yi}$  — координаты вершин заражения.

В следующей строке дано целое число  $Q$  — количество поселений на Мидаве ( $1 \leq Q \leq 10^5$ ).

В следующих  $Q$  строках даны по два целых числа  $t_{xi}$  и  $t_{yi}$  — координаты каждого из поселений.

Все координаты по модулю не превосходят  $10^9$ . Гарантируется, что данный многоугольник выпуклый, а также, что вершины заданы в порядке обхода против часовой стрелки. Гарантируется, что поселения находятся на расстоянии не меньшем  $10^{-6}$  от границы заражения в любой из дней, кроме нулевого.

**Выходные данные**

Выведите  $Q$  целых чисел — последние дни для поселений в порядке ввода.

входные данные
4 1 3 1 1 3 1 3 3 4 2 2 1 2 4 1 6 2
выходные данные
0 0 2 4

В примере второе поселение будет заражено в нулевой день, так как лежит на границе заражения.

6F. Точка в многоугольнике

1 секунда, 256 мегабайт

**Входные данные**

В первой строке содержится три числа —  $N$  ( $3 \leq N \leq 100\,000$ ) и координаты точки. Последующие  $N$  строк содержат координаты углов многоугольника. Координаты — целые, не превосходят  $10^6$  по модулю.

**Выходные данные**

Одна строка YES, если заданная точка содержится в приведённом многоугольнике или на его границе, и NO в противном случае.

входные данные
3 2 3 1 1 10 2 2 8
выходные данные
YES

7A. Диофантово уравнение

0.25 секунд, 256 мегабайт

Даны натуральные числа  $a$ ,  $b$  и  $c$ . Решите в целых числах уравнение  $ax + by = c$ . Среди множества решений следует выбрать такое, где  $x$  имеет наименьшее неотрицательное значение.

**Входные данные**

Первая строка содержит три целых числа  $a$  и  $b$  и  $c$  ( $1 \leq a, b, c \leq 10^9$ ).

**Выходные данные**

Выведите искомые  $x$  и  $y$  через пробел. Если решения не существует, выведите одну строку «Impossible».

входные данные
1 2 3

выходные данные
1 1

входные данные
10 6 8
выходные данные
2 -2

## 7B. Система линейных уравнений

1 секунда, 256 мегабайт

Дана система из двух линейных сравнений:  $x \equiv a \pmod n$  и  $x \equiv b \pmod m$ , где числа  $n$  и  $m$  не обязательно взаимно простые. Решите эту систему или определите, что она не имеет решений.

### Входные данные

В первой строке входного файла записано единственное число  $1 \leq t \leq 100\,000$ . В следующих  $t$  строках содержатся по четыре целых числа  $a, b, n, m$ , задающих одну систему сравнений. Все числа не превосходят по модулю  $10^4$ ,  $n > 1, m > 1$ .

### Выходные данные

Программа должна вывести  $t$  строк, по одной на каждую систему.

В случае, если система не имеет решений, выведите строку «NO» (без кавычек).

В случае, если решение есть, то необходимо вывести слово «YES» (без кавычек) и два таких целых числа  $x_0$  и  $p$ ,  $0 \leq x_0 < p$ , таких, что множество чисел  $x = x_0 + kp$  (где  $k$  — произвольное целое число) является решением данной системы, и все решения системы описаны этой парой чисел.

входные данные
3 3 2 5 9 1 1 5 9 7 13 20 24
выходные данные
YES 38 45 YES 1 45 NO

## 7C. Все обратные по модулю

3.5 секунд, 512 мегабайт

Дано простое число  $p$ . Найдите обратные по модулю  $p$  ко всем числам от 1 до  $p - 1$ .

### Входные данные

Первая строка содержит число  $p$  ( $1 \leq p \leq 10^8$ ).

### Выходные данные

Для каждого числа от 1 до  $p - 1$  требуется посчитать обратное по модулю  $p$ . Так как чисел очень много, сначала выведите сумму обратных для первых 100 чисел по модулю  $p$ , потом для вторых 100 чисел по модулю  $p$ , потом для третьих 100 чисел и так далее. Если  $p - 1$  не делится на 100, последнее из выведенных вами чисел будет состоять из суммы меньше, чем 100 слагаемых.

входные данные
2
выходные данные
1

входные данные
5

<b>выходные данные</b>
0

Обратите внимание, что сумма 100 чисел тоже берется по модулю, так что все числа, которые вы выводите не должны превышать  $p - 1$ .

## 7D. Все обратные по модулю [Мало памяти]

5 секунд, 64 мегабайта

Дано простое число  $p$ . Найдите обратные по модулю  $p$  ко всем числам от 1 до  $p - 1$ .

### Входные данные

Первая строка содержит число  $p$  ( $1 \leq p \leq 10^8$ ).

### Выходные данные

Для каждого числа от 1 до  $p - 1$  требуется посчитать обратное по модулю  $p$ . Так как чисел очень много, сначала выведите сумму обратных для первых 100 чисел по модулю  $p$ , потом для вторых 100 чисел по модулю  $p$ , потом для третьих 100 чисел и так далее. Если  $p - 1$  не делится на 100, последнее из выведенных вами чисел будет состоять из суммы меньше, чем 100 слагаемых.

<b>входные данные</b>
2
<b>выходные данные</b>
1

<b>входные данные</b>
5
<b>выходные данные</b>
0

Обратите внимание, что сумма 100 чисел тоже берется по модулю, так что все числа, которые вы выводите не должны превышать  $p - 1$ .

## 7E. Все обратные по модулю 2

1 секунда, 256 мегабайт

Дано простое число  $p$ . Найдите обратные по модулю  $p$  ко всем числам от 1 до  $p - 1$ .

### Входные данные

Первая строка содержит число  $p$  ( $1 \leq p \leq 10^9$ ).

### Выходные данные

Так как  $n$  очень большое, выведите сумму обратных по модулю  $p$  ко всем числам от 1 до  $p - 1$ , взятую по модулю  $p$ .

<b>входные данные</b>
2
<b>выходные данные</b>
1

<b>входные данные</b>
5
<b>выходные данные</b>
0

Обратите внимание, что сумма чисел тоже берется по модулю, так что число, которое вы выводите не должно превышать  $p - 1$ .

## 7F. Очередь в столовой

2 секунды, 256 мегабайт

Главный распорядитель столовой Галактической Школы Добра Иннокентий очень любит порядок. Но каждый день на Очень Большой Перемене, когда ученики направляются на обед, в его владениях воцаряется хаос.

Начинается всё вполне безобидно — двое самых проворных школьников встают в очередь. Далее очередь расширяется в  $k$  этапов. На  $i$ -м этапе ( $1 \leq i \leq k$ ) в каждый промежуток между соседними школьниками, уже стоящими в очереди, вклинивается по  $a_i$  человек. Например, в случае  $k = 2, a_1 = 3, a_2 = 1$  после первого этапа расширения в очереди оказывается 5 человек, а после второго — 9.

Несмотря на название учебного заведения, такие метаморфозы очереди не проходят без ссор и потасовок. Уставший от бардака Иннокентий твёрдо решил бороться с этим безобразием. Для того чтобы железной рукой наводить порядок, он хочет научиться выяснять, как происходил процесс расширения очереди, зная только итоговое число  $n$  учеников в ней. Понимая, что по  $n$  процесс не восстанавливается однозначно, Иннокентий хочет найти максимально возможное число этапов расширения очереди  $k$ , а также соответствующий ему набор чисел  $a_i$  ( $1 \leq i \leq k$ ), обозначающих количества школьников, которые вклинивались между каждыми двумя соседями в очереди на каждом из этих этапов.

Количество воспитанников Школы, которые могут прийти в столовую, поистине огромно, поэтому за помощью в этом нелёгком деле Иннокентий обратился к вам.

### Входные данные

На вход программе подаётся одно целое число  $n$  ( $3 \leq n \leq 2^{64} - 1$ ) — итоговое число учеников в очереди.

### Выходные данные

В первой строке выведите одно целое положительное число  $k$  — максимальное количество этапов расширения очереди. Во второй строке выведите через пробел  $k$  целых положительных чисел  $a_i$  ( $1 \leq i \leq k$ ). В случае, если удовлетворяющих условию последовательностей  $a_i$  максимальной длины несколько, выведите любую из них.

ВХОДНЫЕ ДАННЫЕ
4
ВЫХОДНЫЕ ДАННЫЕ
1 2

ВХОДНЫЕ ДАННЫЕ
9
ВЫХОДНЫЕ ДАННЫЕ
3 1 1 1

В первом примере, очевидно, есть только одна возможность — на первом шаге вклинивается два школьника.

Во втором примере процесс определён неоднозначно: один вариант развития событий с  $k = 2$  приведён в условии, однако максимально возможное число этапов расширения очереди равно трём.

## 7G. Проверка на простоту

1 секунда, 256 мегабайт

Дано  $n$  натуральных чисел  $a_i$ . Определите для каждого числа, является ли оно простым.

### Входные данные

Программа получает на вход число  $n$ ,  $1 \leq n \leq 1000$  и далее  $n$  чисел  $a_i$ ,  $1 \leq a_i \leq 10^{18}$ .

### Выходные данные

Если число  $a_i$  простое, программа должна вывести YES, для составного числа программа должна вывести NO.

входные данные
4 1 5 10 239
выходные данные
NO YES NO YES

## 7Н. Глина или не глина?

2 секунды, 512 мегабайт

Вам дано положительное целое число  $k$ . Найдите количество троек положительных целых чисел  $(n, p, m)$ , таких что  $n^2 - k \cdot p^m = 1$ , где  $p$  — простое число, либо сообщите, что существует бесконечное количество таких троек чисел.

### Входные данные

В первой строке записано число  $t$  ( $1 \leq t \leq 100$ ) — количество наборов входных данных.

Далее следует описание наборов входных данных.

Единственная строка описания набора входных данных содержит целое число  $k$  ( $1 \leq k \leq 10^9$ ).

### Выходные данные

Для каждого набора входных данных выведите количество троек положительных чисел  $(n, p, m)$ , таких что  $n^2 - k \cdot p^m = 1$  и  $p$  — простое число, либо  $-1$ , если существует бесконечное количество таких троек чисел.

входные данные
2 5 22
выходные данные
3 0

В первом наборе для  $k = 5$  существуют три подходящие тройки чисел:  $(4, 3, 1)$ ,  $(6, 7, 1)$  и  $(9, 2, 4)$ .

Во втором наборе для  $k = 22$  не существует ни одной подходящей тройки чисел.

## 8А. Задача для второклассника

2.5 секунд, 256 мегабайт

Вам даны два числа. Необходимо найти их произведение.

### Входные данные

Входные данные состоят из двух строк, на каждой из которых находится целое одно **целое** число, длина которого не превосходит двухсот пятидесяти тысяч символов.

### Выходные данные

Выведите произведение данных чисел.

входные данные
2 2



выходные данные
4

входные данные
1 -1
выходные данные
-1

входные данные
-1 -239
выходные данные
239

входные данные
0 -1
выходные данные
0

## 8B. ФФТ по известному модулю

2 секунды, 256 мегабайт

Предлагается в этой задаче реализовать перемножение двух многочленов по известному модулю  $MOD = 998\,244\,353$ . Используя написанную вами в этой задаче функцию, вы сможете сдавать без проблем большинство задач по комбинаторике, в которой нужно ФФТ.

### Входные данные

В первой строке находится целое число  $n$  ( $0 \leq n \leq 18$ ).

Во второй строке находится  $2^n$  целых чисел  $a_0, a_1, \dots, a_{2^n-1}$  ( $0 \leq a_i < MOD - 1$ ). Первый многочлен для перемножения это

$$A(x) = \sum_{i=0}^{2^n-1} a_i x^i.$$

Во второй строке находится  $2^n$  целых чисел  $b_0, b_1, \dots, b_{2^n-1}$  ( $0 \leq b_i < MOD - 1$ ). Первый многочлен для перемножения это

$$B(x) = \sum_{i=0}^{2^n-1} b_i x^i.$$

### Выходные данные

Пусть многочлен  $C(x) = A(x) \cdot B(x)$ . Все коэффициенты при перемножении берутся по модулю  $MOD$ . Тогда напишем, что

$$C(x) = \sum_{i=0}^{2^{n+1}-1} c_i x^i.$$

Выведите  $c_0, c_1, \dots, c_{2^{n+1}-1}$ .

входные данные
1 1 1 1 1
выходные данные
1 2 1 0

## 8C. HEX-Hell и сломанная строка

2.0 с, 256 мегабайт

Серёжа потерял место, где в редакторе шестнадцатеричных кодов HEX-Hell находилась его строка. И, так как он большой молодец, саму строку он сломал.

Напоминаем, что один байт — две шестнадцатеричных цифры из диапазона  $[0-9A-F]$ . Последовательность кодов в редакторе в данный момент имеет длину не более 125 000 байт (т.е. в ней не более 250 000 символов, и она имеет четную длину).

А также, у Вас есть серёжина битовая строка из нулей и единиц, «поломанная» в некоторых местах (некоторые биты будут заменены на знаки ?).

Пожалуйста, для каждого возможного начала этой строки в редакторе выведите количество совпадающих нулей и единиц. При этом можно считать, что знаки вопроса совпадают с чем угодно! По этой информации горе-Серёжа разберется со своими строками сам. Только найдите количество совпадений!

**Входные данные**

Во входном файле две непустых строки. Первая состоит из символов от 0 до 9 и от A до F. Количество таких символов чётно и не более 250 000.

Вторая строка состоит из нулей, единиц и знаков вопроса. Её длина кратна восьми (она тоже задаёт последовательность байт) и не превосходит четырех длин первой строки (она помещается в редактор).

**Выходные данные**

Если длина (в символах) первой строки  $n$ , а второй  $k$ , то Вам необходимо вывести в первой строке выходного файла  $n/2 - k/8 + 1$  целых чисел — количество совпадающих бит при прикладывании второй строки к некоторому месту первой строки.

Прикладывания упорядочены естественным образом — слева направо.

входные данные
B2D6 1011?010
выходные данные
8 5

B2D6 в двоичном виде это 10110010 11010110

При прикладывании шаблона 1011?010 к первому байту 10110010 получаем восемь совпадений, а ко второму 11010110 — пять.

Условие  
недоступно  
на  
русском  
языке

8Е. Количество путей

5 секунд, 256 мегабайт

Дано дерево из  $n$  вершин. Для каждого  $d = 1 \dots n - 1$  найдите количество путей длины  $d$ .

**Входные данные**

Первая строка содержит  $n$  ( $1 \leq n \leq 50000$ ) — количество вершин.

Следующие  $n - 1$  строк содержат пары чисел  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ), описывающие рёбра дерева.

**Выходные данные**

Выведите  $n - 1$  число, где  $i$ -е — количество путей длины  $i$ .

входные данные
3 1 2 2 3
выходные данные
2 1

