

## Задача А. Паросочетание

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Двудольным графом называется неориентированный граф  $(V, E)$ ,  $E \subseteq V \times V$  такой, что его множество вершин  $V$  можно разбить на два множества  $A$  и  $B$ , для которых  $\forall (e_1, e_2) \in E$   $e_1 \in A$ ,  $e_2 \in B$  и  $A \cup B = V$ ,  $A \cap B = \emptyset$ .

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор  $S \subseteq E$ , что для любых двух рёбер  $e_1 = (u_1, v_1)$ ,  $e_2 = (u_2, v_2)$  из  $S$   $u_1 \neq u_2$  и  $v_1 \neq v_2$ .

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 250$ ), где  $n$  — число вершин в множестве  $A$ , а  $m$  — число вершин в  $B$ .

Далее следуют  $n$  строк с описаниями рёбер —  $i$ -я вершина из  $A$  описана в  $(i + 1)$ -й строке файла. Каждая из этих строк содержит номера вершин из  $B$ , соединённых с  $i$ -й вершиной  $A$ . Гарантируется, что в графе нет кратных ребер. Вершины в  $A$  и  $B$  нумеруются независимо (с единицы). Список завершается числом 0.

### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число  $l$  — количество рёбер в максимальном паросочетании.

### Пример

стандартный ввод	стандартный вывод
2 2 1 2 0 2 0	2

## Задача В. Замощение доминошками

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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.

### Формат входных данных

Первая строка входного файла содержит 4 целых числа  $n, m, a, b$  ( $1 \leq n, m \leq 100, |a| \leq 1000, |b| \leq 1000$ ). Каждая из последующих  $n$  строк содержит по  $m$  символов: символ '.' (точка) обозначает занятую клетку поля, а символ '\*' (звёздочка) — свободную.

### Формат выходных данных

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

### Пример

стандартный ввод	стандартный вывод
2 3 3 2 .** .*.	5

## Задача С. Проблема падишаха

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Мудрый падишах внимательно следит за благополучием своих подданных, когда вершит их судьбы. В частности, на нем все заботы о вступающих в брачный возраст юношах и девушках его страны. И, как положено серьезному правителю, все по науке — перед тем, как творить молодые семьи, падишах провел Глобальное тестирование и по 100-балльной шкале определил совместимость всех юношей и девушек в совместном браке.

А дальше что? Падишах наслышан про задачу о назначении, но ему не нравится ее установка. Действительно, может ли быть спокойна его душа даже в случае всеобщего благополучия, если кому-то из подданных плохо? И можно ли жертвовать интересами хотя бы одной семьи во благо общества? Конечно, нет!

Падишаху милее другая мысль. Он хочет создать максимальное число семей, причем сделать это таким образом, чтобы минимальная совместимость в семье была максимальной. А решить эту неклассическую задачу он просит вас. Помогите падишаху!

### Формат входных данных

В первой строке входных данных содержатся два целых числа  $n$  и  $m$  — количество юношей и количество девушек соответственно ( $1 \leq n, m \leq 200$ ). Последующие  $n$  строк содержат по  $m$  целых чисел от 0 до  $10^9$  — коэффициент совместимости соответствующей пары (меньшее значение менее способствует супружеской жизни).

### Формат выходных данных

В единственную строку выходного файла выведите наименьший искомый балл, при котором возможно создание максимально возможного количества семейных пар.

### Пример

стандартный ввод	стандартный вывод
3 4 77 88 31 67 96 30 2 68 35 39 76 45	76

## Задача D. Разбиение на пути

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Разбейте рёбра неориентированного графа на минимальное количество рёберно-простых путей.

### Формат входных данных

Дан граф. На первой строке число вершин  $n$  ( $1 \leq n \leq 20\,000$ ) и число рёбер  $m$  ( $1 \leq m \leq 20\,000$ ).  
Следующие  $m$  строк содержат описание рёбер графа. Каждая строка по два числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ). Между каждыми двумя вершинами не более одного ребра. Граф связан.

### Формат выходных данных

На первой строке минимальное число путей. На каждой следующей строке описание очередного пути – номера вершин в порядке прохождения.

### Пример

стандартный ввод	стандартный вывод
7 7	3
1 2	2 4 7 6
4 1	7 5
6 7	4 1 2 3
5 7	
7 4	
2 3	
4 2	

## Задача Е. Наркоконтроль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В маленьком городке «М» начала действовать служба наркоконтроля. Первая задача службы — выяснить, сколько наркоторговцев работает в окрестности города. Агенты службы опросили всех наркозависимых в городе и составили список случаев продажи травки, произошедших за одни сутки, с указанием места и времени наблюдения. Теперь аналитики хотят понять, сколько же на самом деле есть наркоторговцев. Из данных разведки известна максимальная скорость, с которой может двигаться наркоторговец. Аналитики просят вас узнать, какое минимальное количество наркоторговцев могли участвовать во всех зафиксированных случаях продажи травки.

### Формат входных данных

На первой строке входного файла содержатся целые числа  $n$  и  $v$  — количество случаев продажи травки и максимальная скорость наркоторговца ( $1 \leq n \leq 100, 1 \leq v \leq 10000$ ). Следующие  $n$  строк содержат описания случаев продажи травки в формате «ЧЧ:ММ  $x$   $y$ », где ЧЧ:ММ — время продажи,  $x$  и  $y$  — координаты места, в котором продавалась травка (для простоты будем считать, что всё происходило на плоскости). Координаты по модулю не превышают 1000. Скорость выражена в км/ч, координаты — в км.

### Формат выходных данных

Выведите в выходной файл одно число — минимальное возможное количество наркоторговцев.

### Пример

стандартный ввод	стандартный вывод
4 1 12:00 0 0 13:10 0 1 14:00 1 0 15:00 1 1	2

## Задача F. Эйлеров путь

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

Дан неориентированный связный граф, не более трех вершин имеет нечетную степень. Требуется определить, существует ли в нем путь, проходящий по всем ребрам.

Если такой путь существует, необходимо его вывести.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  — количество вершин графа ( $1 \leq n \leq 100\,000$ ). Далее следуют  $n$  строк, задающих ребра. В  $i$ -й из этих строк находится число  $m_i$  — количество ребер, инцидентных вершине  $i$ . Далее следуют  $m_i$  натуральных чисел — номера вершин, в которые ведут ребро из  $i$ -й вершины.

Граф может содержать кратные ребра, но не содержит петель.

Граф содержит не более 300 000 ребер.

### Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число  $k$  — количество ребер в искомом маршруте, а во вторую  $k + 1$  число — номера вершин в порядке их посещения.

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

Если решений несколько, выведите любое.

### Пример

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

## Задача G. Таня и пароль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Пока папа был на работе, маленькая девочка Таня решила поиграть с папиным паролем к секретной базе данных. Папин пароль представляет собой строку, состоящую из  $n + 2$  символов. Она выписала все возможные  $n$  трёхбуквенных подстрок пароля на бумажки, по одной на каждую бумажку, а сам пароль выкинула. Каждая трёхбуквенная подстрока была выписана на бумажки столько раз, сколько она встречалась в пароле. Таким образом, в итоге у Тани оказалось  $n$  бумажек.

Потом Таня поняла, что папа расстроится, если узнает о ее игре, и решила восстановить пароль или, по крайней мере, хотя бы какую-то строку, соответствующую получившемуся набору трёхбуквенных строк. Вам предстоит помочь ей в этой непростой задаче. Известно, что папин пароль состоял из строчных и заглавных букв латинского алфавита, а также из цифр. Строчные и заглавные буквы латинского алфавита считаются различными.

### Формат входных данных

В первой строке следует целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ), количество трёхбуквенных подстрок, которые получились у Тани.

Следующие  $n$  строк каждая содержат по три буквы, образующие подстроку пароля папы. Каждый символ во вводе — строчная или заглавная буква латинского алфавита или цифра.

### Формат выходных данных

Если во время игры Таня что-то напутала, и строк, соответствующих данному набору подстрок, не существует, то выведите «NO».

Если же возможно восстановить строку, соответствующую данному набору подстрок, то выведите «YES», а затем любой подходящий вариант пароля.

### Примеры

стандартный ввод	стандартный вывод
5 aca aba aba cab bac	YES abacaba
4 abc bCb cb1 b13	NO
7 aaa aaa aaa aaa aaa aaa aaa	YES aaaaaaaaa

## Задача Н. Максимальное паросочетание

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3.5 секунд  
Ограничение по памяти: 256 мегабайт

Дан двудольный граф. У каждой вершины графа есть вес. Вес ребра — сумма весов его концов. Вес паросочетания — сумма весов рёбер, входящих в паросочетание. Нужно найти паросочетание максимального веса. Заметим, это паросочетание может содержать сколько угодно рёбер, единственное условие — вес паросочетания должен быть максимальным.

Напомним, что паросочетанием в двудольном графе называется набор рёбер этого графа такой, что никакие два ребра набора не имеют общих вершин.

### Формат входных данных

В первой строке заданы размеры долей  $n$  и  $m$  ( $1 \leq n, m \leq 5000$ ) и количество рёбер  $e$  ( $0 \leq e \leq 10000$ ). Вторая строка содержит  $n$  целых чисел от 0 до 10000 — веса вершин первой доли. Третья строка содержит  $m$  целых чисел от 0 до 10000 — веса вершин второй доли. Следующие  $e$  строк содержат рёбра графа. Каждое ребро описывается парой целых чисел  $a_i b_i$ , где  $1 \leq a_i \leq n$  — номер вершины первой доли и  $1 \leq b_i \leq m$  — номер вершины второй доли.

### Формат выходных данных

В первой строке выведите  $w$  — максимальный вес паросочетания. Во второй строке выведите  $k$  — количество рёбер в паросочетании максимального веса. В следующей строке выведите  $k$  различных чисел от 1 до  $e$  — номера рёбер в паросочетании. Если максимальных по весу паросочетаний несколько, разрешается вывести одно любое.

### Примеры

стандартный ввод	стандартный вывод
4 3 3 2 0 9 9 1 0 9 1 2 2 1 1 1	3 2 1 2
3 2 4 1 2 3 1 2 1 1 2 1 2 2 3 2	8 2 2 4



## Задача I. Xor the Graph

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф, состоящий из  $N$  вершин и  $M$  рёбер. Каждому ребру присвоено значение, 0 или 1. Операция состоит в том, чтобы выбрать путь (не обязательно простой), и изменить значения, записанные на каждом его ребре на противоположные. Сделайте значения на всех рёбрах нулями за минимальное число операций.

### Формат входных данных

На первой строке заданы два целых числа  $N$  и  $M$  ( $1 \leq N \leq 10^5$ ,  $0 \leq M \leq 10^5$ ).

В каждой из следующих  $M$  строк записаны по три целых числа  $a$ ,  $b$  и  $c$ . Они означают, что существует ребро между вершинами  $(a, b)$ , изначальное значение которого равно  $c$  ( $1 \leq a, b \leq n$ ,  $0 \leq c \leq 1$ ).

### Формат выходных данных

Выведите количество операций  $K$  на первой строке.

В последующих  $K$  строках выведите пути, которые вы выбрали:

- Первым числом выведите  $P$  — количество вершин в пути
- Затем выведите  $P$  целых чисел  $v_1, v_2, \dots, v_P$  — индексы вершин. Для всех  $1 \leq i \leq P - 1$   $(v_i, v_{i+1})$  должно являться ребром в заданном графе.

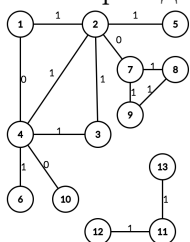
Решение будет считаться верным, если оно использует минимальное количество операций, и при этом  $\sum P \leq 4 * M$ .

### Пример

стандартный ввод	стандартный вывод
13 14	3
1 2 1	5 1 2 3 4 6
2 3 1	8 4 2 7 8 9 7 2 5
3 4 1	3 12 11 13
2 4 1	
1 4 0	
4 6 1	
4 10 0	
2 5 1	
2 7 0	
7 8 1	
8 9 1	
9 7 1	
11 12 1	
11 13 1	

### Замечание

Ниже приведён граф в первом тесте.



## Задача J. Тимбилдинг

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Начался новый учебный год, и в университет Берляндии пришли  $n$  новых студентов, разбитых на  $k$  групп, некоторые из которых могут быть пустыми. Среди студентов есть  $m$  пар знакомых, причём знакомые студенты могут быть как из одной, так и из разных групп. Алиса, как куратор нового набора, позвала всех новоприбывших на организационную встречу. На ней она хочет устроить игру, чтобы еще незнакомые студенты получше узнали друг друга. Для этого она выберет две группы, студенты из которых будут играть. При этом правила игры требуют разделить участников на две команды так, чтобы внутри каждой из команд никто не знал друг друга.

Алису интересует: сколько существует способов выбрать две различные группы студентов так, чтобы получилось сыграть в игру по всем правилам. При этом в игре должны участвовать все студенты обеих групп.

Обратите внимание, что команды, на которые Алиса разделит студентов, не обязаны совпадать с группами, в которых учатся участники.

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq n \leq 500\,000$ ,  $0 \leq m \leq 500\,000$ ,  $2 \leq k \leq 500\,000$ ) — количество студентов, количество пар знакомых студентов и количество групп, соответственно.

Во второй строке заданы  $n$  целых чисел  $c_i$  ( $1 \leq c_i \leq k$ ), где  $c_i$  равняется номеру группы, в которой учится  $i$ -й студент.

Далее следуют  $m$  строк. В  $i$ -й строке записаны два числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ), означающие, что  $a_i$ -й и  $b_i$ -й студент знают друг друга. Гарантируется, что  $a_i \neq b_i$ , и что если пара студентов знает друг друга, то она встречается ровно один раз.

### Формат выходных данных

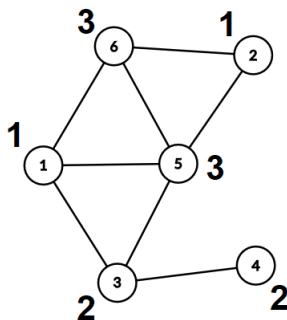
Выведите одно число — количество способов выбрать две различные группы так, чтобы получилось сыграть в игру по всем правилам.

## Примеры

стандартный ввод	стандартный вывод
6 8 3 1 1 2 2 3 3 1 3 1 5 1 6 2 5 2 6 3 4 3 5 5 6	2
4 3 3 1 1 2 2 1 2 2 3 3 4	3
4 4 2 1 1 1 2 1 2 2 3 3 1 1 4	0
5 5 2 1 2 1 2 1 1 2 2 3 3 4 4 5 5 1	0

## Замечание

Граф знакомств для первого тестового примера выглядит следующим образом (рядом с каждым студентом подписан номер его группы):



В данном случае нам подходят способы:

- Выбрать первую и вторую группу — например, можно отнести студентов номер 1 и 4 к первой команде, а студентов номер 2 и 3 ко второй.
- Выбрать вторую и третью группу — можно отнести студентов номер 3 и 6 к первой команде, а студентов номер 4 и 5 ко второй.

- Выбрать первую и третью группы мы не можем, потому что не существует разбиения на команды, удовлетворяющего правилам игры.

Во втором тестовом примере мы можем выбрать любую пару групп. Обратите внимание, что несмотря на то, что в третьей группе нет студентов, мы все равно можем ее выбирать.

## Задача К. Антенна

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Для связи с Землёй членам экспедиции на Марс необходимо собрать антенну. Антенна в разобранном состоянии представляет собой  $n$  фрагментов,  $i$ -й фрагмент представляет собой штангу длиной  $s_i$  сантиметров, на которой закреплены  $m_i$  перекладин. Каждый фрагмент содержит хотя бы одну перекладину.

У каждой штанги есть начало, в котором расположен штекер, и конец, в котором расположено гнездо. Любые две штанги можно последовательно соединить, присоединив начало одной к концу другой. Для каждой перекладины известно расстояние от начала её штанги в сантиметрах. Для  $i$ -го фрагмента это расстояние может быть от 0 до  $s_i$ , значение 0 означает, что перекладина находится непосредственно в начале штанги, значение  $s_i$  — что она находится непосредственно в конце штанги. Толщиной перекладин и размерами штекера и гнезда следует пренебречь.

Чтобы корректно собрать антенну, необходимо соединить в некотором порядке все  $n$  фрагментов, при этом расстояние между любыми двумя соседними перекладинами должно быть одинаковым.

К сожалению, члены экспедиции забыли инструкцию по сборке антенны на Земле, а передать её на Марс не представляется возможным — ведь антенна ещё не собрана. Помогите исследователям!

Требуется определить, в каком порядке необходимо соединить фрагменты антенны, чтобы установить связь с Землей.

### Формат входных данных

В первой строке дано одно число  $n$  — количество фрагментов ( $1 \leq n \leq 100\,000$ ).

Далее дано описание  $n$  фрагментов. В первой строке описания фрагмента даны два целых числа  $m_i$  и  $s_i$  — количество перекладин и длина штанги в  $i$ -м фрагменте ( $1 \leq m_i \leq 100\,000$ ,  $0 \leq s_i \leq 10^9$ ). В следующей строке даны  $m_i$  целых чисел  $p_{i,j}$  — позиции перекладин,  $p_{i,j}$  равно расстоянию в сантиметрах от начала штанги до  $j$ -й перекладины на ней ( $0 \leq p_{i,1} < p_{i,2} < \dots < p_{i,m_i} \leq s_i$ ).

Сумма всех  $m_i$  не превышает 100 000.

### Формат выходных данных

Если собрать антенну указанным образом возможно, в первой строке выведите «Yes», а во второй строке выведите перестановку чисел от 1 до  $n$  — номера фрагментов в порядке, в котором их следует соединить, начало каждого следующего фрагмента в этом порядке присоединяется к концу предыдущего фрагмента. Если существует несколько подходящих ответов, можно вывести любой из них.

Если собрать антенну невозможно, в единственной строке выведите «No».

## Система оценки

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

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	8	$n \leq 8, m_i = 1, s_i \leq 100$		первая ошибка
2	8	$n \leq 8, s_i \leq 100$	1	первая ошибка
3	21	$n \leq 1000$	1, 2	первая ошибка
4	21	$\sum m_i > n$		первая ошибка
5	21	$s_i \leq 100$	1, 2	первая ошибка
6	21	нет	1-5	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
3 1 7 3 1 8 6 2 8 1 6	Yes 2 1 3
1 1 7 5	Yes 1
1 3 10 2 5 9	No
3 1 5 3 1 3 3 1 6 3	No
4 1 5 0 1 0 0 1 3 3 1 0 0	Yes 3 2 4 1

## Задача L. Футболки на олимпиаду

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Наконец-то пришло время для финала Закрытой олимпиады школьников по программированию! Все отборы пройдены, сувенирные футболки лежат в пакетах. Вот только распределение участников по площадкам еще не спланировано.

У жюри олимпиады есть  $m$  площадок проведения. На  $i$ -й площадке лежит пакет, в котором находится  $a_i$  футболок. Это значит, что на эту площадку можно зарегистрировать не более  $a_i$  участников. Всего же среди финалистов будет  $n$  человек.

Финал Закрытой олимпиады школьников по программированию проходит в городе-герое Кленинграде. Как любой другой город Флатландии, он расположен в декартовой системе координат. Жюри знает адрес каждого финалиста — координаты  $X_i, Y_i$ , а также адреса площадок — координаты  $x_i, y_i$ .

Жюри хочет так распределить участников по точкам проведения, чтобы всем участникам хватило футболок, при этом максимальное из расстояний, которое придется пройти участнику до точки проведения, было как можно меньше. Поскольку жюри олимпиады сейчас везет на точки проведения другую сувенирку, Вас попросили провести распределение участников по площадкам.

### Формат входных данных

В первой строке заданы два целых числа  $n, m$  ( $1 \leq n, m \leq 500$ ) — количество участников олимпиады и точек проведения.

В следующих  $n$  строках заданы пары целых чисел  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^6$ ) — координаты  $i$ -го участника олимпиады.

В следующих  $m$  строках заданы тройки целых положительных чисел  $X_i, Y_i, a_i$  ( $1 \leq X_i, Y_i \leq 10^6, 1 \leq a_i \leq n$ ) — координаты  $i$ -й площадки и ее вместимость.

Гарантируется, что сумма  $a_i$  не меньше  $n$ , а так же что она не превосходит 1000.

### Формат выходных данных

Выведите одно число — максимальное расстояние, которое придется пройти участнику до точки проведения олимпиады при оптимальном распределении. Ответ будет считаться верным, если его абсолютная или относительная ошибка не будет превосходить  $10^{-6}$ . Формально, пусть ваш ответ равен  $a$ , а ответ жюри —  $b$ . Ваш ответ считается правильным, если  $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$ .

### Примеры

стандартный ввод	стандартный вывод
3 2 1 1 2 3 3 2 1 1 1 2 2 2	1.0000000000
3 2 100 100 101 101 102 102 101 101 2 105 105 2	4.2426406871

### Замечание

В первом примере надо отправить первого участника на первую площадку, а оставшихся двух — на вторую. Тогда первый участник пройдет расстояние 0, а второй и третий — по 1.

Во втором примере надо первого и второго участников отправить на площадку 1, а третьего — на площадку 2. Тогда первый участник пройдёт расстояние 1.41421356237, второй — 0, а третий — 4.2426406871.

## Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		$n$	$m$	$a_i$		
0	0	–	–	–	–	Тесты из условия.
1	10	–	–	$a_i = n$	–	
2	11	–	$m \leq 2$	–	0	
3	15	–	$m \leq 3$	–	0, 2	
4	14	$n \leq 10$	$m \leq 10$	–	0	
5	13	$n \leq 50$	$m \leq 50$	–	0, 4	
6	7	$n \leq 200$	$m \leq 200$	–	0, 4, 5	
7	9	$n \leq 300$	$m \leq 300$	–	0, 4, 5, 6	
8	10	$n \leq 400$	$m \leq 400$	–	0, 4, 5, 6, 7	<b>Offline-проверка.</b>
9	11	–	–	–	0–8	<b>Offline-проверка.</b>