

Задача А. Инопланетное вторжение

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

Недавно на Землю приземлился инопланетный космический корабль, вооружённый по последнему слову техники. Он немедленно окружил себя непробиваемым защитным полем, так что теперь он неуязвим даже для прямого ядерного удара. Однако нашим разведчикам удалось разузнать, что по прибытии из корабля вышли несколько инопланетян, некоторые из которых — красные, некоторые — зелёные, а некоторые — синие. Инопланетяне встали по кругу, после чего стали происходить странные вещи. А именно, два соседних инопланетянина разных цветов иногда сливались в единое целое, которое по истечении некоторого времени превращалось в инопланетянина третьего цвета (таким образом в круге становилось на одного инопланетянина меньше).

Наши разведчики догадались, что инопланетян разных цветов раздрают непреодолимые противоречия, и для того, чтобы напасть (или, может, предложить руку дружбы?) им нужно образовать круг из инопланетян одного цвета. Для того, чтобы знать, к чему готовиться, нашим правительственным службам требуется узнать минимальное и максимальное возможное количество инопланетян в круге по достижении этой цели.

Вас наняли для этой ответственной работы. Напишите программу, которая определила бы минимальное и максимальное количество инопланетян в момент, когда все они станут одноцветными.

Формат входных данных

В первой строке входного файла задано текущее количество инопланетян n ($1 \leq n \leq 3\,000\,000$). Вторая строка имеет длину n символов, каждый из которых — либо 'R', если соответствующий инопланетянин красный, либо 'G' — если зелёный, либо 'B' — если синий.

Формат выходных данных

Выведите через пробел два числа — минимальное и максимальное возможное количество одноцветных инопланетян по завершении процесса слияния.

Примеры

стандартный ввод	стандартный вывод
4 BGRR	1 3
6 BBRRGG	2 2

Замечание

Если в первом примере первый инопланетянин сначала сольётся со вторым, то на Землю вторгнутся три инопланетянина красного цвета. Если же, к примеру, сначала сольются

первый и четвёртый инопланетянин (таким образом получится круг GRG), потом второй и третий (получится круг GB), а в оставшемся круге сольются два оставшихся инопланетянина, то к вторжению останется всего лишь один красный инопланетянин.

Задача В. Коробки

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

Дано N коробок в пространстве. Про каждую коробку известны её три измерения. Коробки можно вращать — то есть выбирать, какое измерение будет длиной, какое — шириной, а какое — высотой. Из этих коробок необходимо построить башню максимальной высоты. Коробки разрешается ставить друг на друга, только если длина коробки сверху не превосходит длины коробки снизу, а ширина коробки сверху — ширины коробки снизу.

Формат входных данных

В первой строке задано целое число N ($1 \leq N \leq 30$). Следующие N строк содержат по три целых числа a_i , b_i и c_i ($1 \leq a_i, b_i, c_i \leq 10^6$) — три измерения i -ой коробки.

Формат выходных данных

В первой строке выведите максимальную высоту башни H . Во второй строке выведите целое число k — количество коробок в искомой башне. В последующих k строках выведите по четыре числа для каждой коробки — её номер m_i , длину l_i , ширину d_i и высоту h_i (коробки нужно выводить в порядке снизу вверх). Коробки нумеруются с единицы в порядке, заданном во входном файле.

Пример

стандартный ввод	стандартный вывод
3	5
3 1 3	2
2 2 2	1 1 3 3
1 2 1	3 1 1 2

Замечание

Решение, работающее для $N \leq 20$, будет набирать не менее 70 баллов.

Задача С. Метро

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

Метро города Нью-Славец состоит из n станций. Расположение каждой станции можно задать координатами в декартовой системе координат: x_i , y_i , z_i . При создании подземки

для помощи привлекли учёных с дружественной нам звезды. Инопланетяне использовали неизвестные технологии и создали очень сложную систему дорог. Чтобы добраться от станции i до станции j , требуется $\rho(i, j) = |x_i - y_j| + |y_i - z_j| + |z_i - x_j|$ миллисекунд. Учёных со всей Земли особенно удивляет, что $\rho(i, i)$ может быть больше нуля и $\rho(i, j)$ может быть не равно $\rho(j, i)$.

Нью-Славцы пользуются метро уже несколько лет, но так и не поняли, сколько времени добираться между станциями. Известная рок-группа «Високосная секунда» решила написать новую песню про две самые удалённые станции метро. Помогите группе порадовать слушателей новой песней, найдите номера двух самых удалённых станций.

Стоит заметить, что в метро нельзя делать пересадки, и группу интересуют две различные станции.

Формат входных данных

В первой строке задано натуральное число n — количество станций в Нью-Славском метро ($2 \leq n \leq 1\,000\,000$). Далее на n различных строках записаны по три числа — координаты каждой станции x_i, y_i, z_i . Координаты по модулю не превосходят 10^5 .

Формат выходных данных

В первой строке выведите время в миллисекундах между двумя наиболее удалёнными станциями, то есть максимальное $\rho(x, y)$ ($x \neq y$). Во второй строке выведите два различных числа x и y — номера двух наиболее удалённых станций. Станции нумеруются с единицы в порядке задания их во входном файле. Если ответов несколько — выведите любой.

Примеры

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

Задача D. Два остовных дерева

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

Задан неориентированный граф, рёбра которого можно разбить на два непересекающихся остовных дерева. Вам необходимо найти одно из таких разбиений.

Формат входных данных

Входные файлы к этой задаче являются открытыми; их можно найти по адресу <http://192.168.21.239/twost/>.

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и рёбер в графе. Следующие m строк содержат описания рёбер графа. Каждое ребро задаётся номерами концов. Гарантируется, что в графе нет петель, но могут быть кратные рёбра.

Вершины и рёбра графа нумеруются с единицы.

Формат выходных данных

Решением задачи считается набор выходных файлов, соответствующих входным, запакованный архиватором `gzip` и утилитой `tar` (команда `'tar -czf archive.tar.gz ???.out'` создаёт `gzip`-архив `'archive.tar.gz'`, содержащий все файлы, подходящие под шаблон `'???.out'`). Архив должен содержать по одному файлу `XX.out` на каждый входной файл `XX` (без расширения) или на любую часть этих файлов, и не должен содержать никаких других файлов и папок.

Каждый выходной файл должен содержать искомое разбиение рёбер графа. В первой строке выведите номера рёбер, которые войдут в первое остовное дерево, во второй — номера рёбер, которые войдут во второе остовное дерево. Каждое ребро графа должно появиться ровно в одной из этих двух строк.

При отсылке производится проверка на соответствие всех файлов в архиве формату вывода, и результат сообщается участнику. Возможные результаты проверки следующие:

- **Compilation Error** — в архиве отсутствуют файлы, они находятся не в корневом каталоге архива, либо архив повреждён.
- **Presentation Error** — грубое нарушение формата вывода в одном из тестов.
- **Accepted** — формат вывода всех включённых в архив ответов соответствует условию задачи.

Оценка решения участника производится по последнему посланному архиву с ответами, не получившему **Compilation Error**.

Пример

стандартный ввод	стандартный вывод
3 4 1 2 2 3 3 1 1 2	1 2 3 4