

Задача А. Инопланетные мелодии

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Инопланетянин записывает человеческие мелодии с помощью инструмента, состоящего из 4 клавиш с различными высотами звука. Человеческая мелодия состоит из k нот с высотами a_1, a_2, \dots, a_k .

Инопланетянин хочет сопоставить каждой ноте a_i клавишу пианино так, чтобы выполнялись следующие условия:

- первая нота может быть сопоставлена с любой клавишей;
- если нота a_i выше предыдущей соседней a_{i-1} , то соответствующая клавиша должна иметь более высокий тон, чем клавиша предыдущей ноты;
- если нота a_i ниже предыдущей соседней a_{i-1} , то соответствующая клавиша должна иметь более низкий тон, чем клавиша предыдущей ноты;
- если соседние ноты a_i и a_{i-1} имеют одинаковую высоту, то они должны быть сопоставлены с одной и той же клавишей.

Если вышеописанные условия невозможно выполнить (например, если требуется пятая, более высокая или более низкая клавиша), то инопланетянин вынужден нарушить одно из правил. Требуется определить минимальное количество нарушений, необходимых для корректного отображения всей мелодии.

Формат входных данных

Первая строка содержит одно целое число k ($1 \leq k \leq 10^5$) — количество нот в мелодии; Вторая строка содержит k целых чисел a_1, a_2, \dots, a_k ($1 \leq a_i \leq 10^6$) — высоты нот.

Формат выходных данных

Выведите одно число — минимальное количество нарушений, необходимых для данной мелодии.

Примеры

стандартный ввод	стандартный вывод
5 1 5 31 179 1	0
8 1 2 3 4 5 6 7 8	1

Задача В. Пропуск КП

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Дана последовательность из n контрольных пунктов, которые необходимо посетить по порядку.

Каждый контрольный пункт i имеет координаты (x_i, y_i) . Расстояние между двумя пунктами (x_1, y_1) и (x_2, y_2) определяется как

$$|x_1 - x_2| + |y_1 - y_2|.$$

Необходимо обрабатывать два типа операций:

- **U i x y** — обновить координаты i -го пункта: $(x_i, y_i) := (x, y)$.
- **Q i j** — вычислить минимальное возможное расстояние маршрута от пункта i до пункта j ($i \leq j$), если разрешено пропустить ровно один пункт между ними (но не i и не j). Если длина маршрута равна $j - i + 1 \leq 2$, пропуск невозможен.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 100\,000$) — количество контрольных пунктов и количество запросов.

Далее следуют n строк, каждая из которых содержит два целых числа x_i и y_i ($-1000 \leq x_i, y_i \leq 1000$) — координаты контрольных пунктов в порядке прохождения маршрута.

Далее следуют q строк, каждая из которых описывает один запрос.

- Если строка начинается с символа «U», то за ним следуют три целых числа i , x и y ($1 \leq i \leq n$, $-1000 \leq x, y \leq 1000$). Это означает, что координаты контрольного пункта с номером i изменяются на (x, y) .
- Если строка начинается с символа «Q», то за ним следуют два целых числа i и j ($1 \leq i < j \leq n$).

Формат выходных данных

Для каждого запроса типа «Q» выведите одно целое число — минимальное возможное расстояние между пунктами i и j с учётом возможного пропуска одного внутреннего пункта.

Пример

стандартный ввод	стандартный вывод
5 5	11
-4 4	8
-5 -3	8
-1 5	
-3 4	
0 5	
Q 1 5	
U 4 0 1	
U 4 -1 1	
Q 2 4	
Q 1 4	

Задача С. Последовательное разбиение

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Заданы целые числа n и l , а также последовательность целых чисел a_1, a_2, \dots, a_n .

Рассмотрим процесс, начинающийся с некоторого индекса x ($1 \leq x \leq n$). Рассмотрим последовательность a_x, a_{x+1}, \dots, a_n . Эта последовательность разбивается на минимальное возможное количество непустых подряд идущих отрезков (групп), так что сумма элементов в каждой группе не превосходит l .

Разбиение выполняется последовательно слева направо: при добавлении очередного элемента a_i в текущую группу, если сумма элементов в группе после добавления превысит l , текущая группа завершается, и a_i становится первым элементом новой группы. После добавления последнего элемента a_n последняя группа также завершается.

Для каждого x ($1 \leq x \leq n$) требуется определить:

- количество получившихся групп;
- сумму элементов последней группы.

Формат входных данных

Первая строка содержит два целых числа n и l ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq l \leq 10^9$).

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq l$).

Формат выходных данных

Выполните n строк. В i -й строке выведите два целых числа — количество групп и сумму элементов последней группы, если процесс начинается с $x = i$.

Пример

стандартный ввод	стандартный вывод
5 10	3 2
6 3 8 1 2	3 2
	2 2
	1 3
	1 2

Задача D. Путь по кампусу ФПМИ

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Сережа устал от бесконечных книг и решил прогуляться по кампусу ФПМИ вместе с друзьями, хотя особого интереса к экскурсиям не испытывает. Пока его друзья в восторге исследуют корпуса и лаборатории, Сережа сидит на скамейке у главного входа и размышляет о возможных маршрутах прогулки по кампусу.

Кампус ФПМИ можно представить как таблицу размером r строк на c столбцов. Экскурсия должна начинаться в левом верхнем углу (главный вход) и заканчиваться в правом нижнем углу (выход к метро). Каждая локация (клетка) может быть посещена не более одного раза, но посещать все локации не обязательно. Перемещаться можно только в соседние клетки — вверх, вниз, влево или вправо.

Каждая локация имеет положительное целое число — степень интересности (например, кофейня, вид на башню, лекционный зал и т.п.). Общая интересность маршрута — это сумма значений всех посещённых локаций. Помогите Сереже найти один из самых интересных маршрутов (с максимальной суммой).

Формат входных данных

В первой строке входных данных содержатся два целых числа r и c ($2 \leq r, c \leq 1000$) — размеры таблицы.

В следующих r строках по c положительных целых чисел, не превышающих 1000, — значения интересности соответствующих локаций кампуса.

Формат выходных данных

Выполните единственную строку без пробелов — последовательность букв, описывающую маршрут от левого верхнего до правого нижнего угла. Направления обозначаются буквами: вверх — U, вправо — R, вниз — D, влево — L. Достаточно вывести любой маршрут, достигающий максимальной суммы.

Пример

стандартный ввод	стандартный вывод
2 2 2 15 31 4	DR

Задача Е. Взвешенная точка с манхэттенской окружностью

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На плоскости расположено n точек. Для каждой точки i заданы её координаты (x_i, y_i) и значение g_i .

Необходимо выбрать такую точку на плоскости (возможно, с нецелыми координатами), чтобы сумма значений g_i для всех точек, находящихся на расстоянии не более k , была максимальной.

Расстояние между точками (x_1, y_1) и (x_2, y_2) определяется как $|x_1 - x_2| + |y_1 - y_2|$.

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq n \leq 100\,000$, $1 \leq k \leq 2\,000\,000$) — количество точек и максимальное расстояние.

Следующие n строк содержат по три целых числа g_i , x_i и y_i ($1 \leq g_i \leq 10\,000$, $0 \leq x_i, y_i \leq 1\,000\,000$) — значение и координаты i -й точки.

Формат выходных данных

Выведите одно целое число — максимальную возможную сумму значений g_i для точек, находящихся на расстоянии не более k от некоторой точки на плоскости.

Пример

стандартный ввод	стандартный вывод
5 4	
5 0 0	
2 3 1	
8 5 2	
3 7 3	
6 9 5	17

Замечание

Если выбрать точку $(7, 4)$, то в радиусе $k = 4$ окажутся точки с координатами $(5, 2)$, $(7, 3)$ и $(9, 5)$, чьи значения дают сумму $8 + 3 + 6 = 17$. Это максимальная возможная сумма.

Задача F. Построй путь для каравана

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Маша управляет своим караваном в стратегической игре по освоению пустынной равнины. Её караван состоит из n вагонов, и она хочет проложить путь между двумя оазисами, чтобы перемещаться безопасно и быстро.

Игра происходит на бесконечной сетке. Каждый оазис представлен отрезком с целыми координатами концов, длина которого равна n и который параллелен одной из осей координат. Путь между оазисами должен быть ломаной линией, звенья которой либо параллельны осям, либо проходят под углом 45° . Концы ломаной должны совпадать с концами оазисов (любой конец одного оазиса соединяется с любым концом другого). Путь может пересекать себя или оазисы — в этих местах можно построить мосты или виадуки.

Для безопасного движения каравана все углы ломаной должны быть 135° . Прямые (90°) и острые (45°) углы недопустимы. Также запрещены такие углы между оазисом и первым или последним сегментом пути. Каждый сегмент пути должен иметь длину не меньше n .

Маша хочет построить самый короткий возможный путь. Помогите ей составить его.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 10$) — длину каравана.

Следующие две строки содержат по четыре целых числа каждая: x_1, y_1, x_2, y_2 — координаты концов первого и второго оазисов соответственно. Координаты целые, $0 \leq x_i, y_i \leq 50$. Отрезки оазисов параллельны одной из координатных осей и не пересекаются.

Формат выходных данных

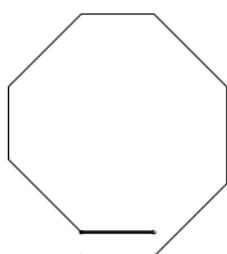
В первой строке выведите одно целое число l ($1 \leq l \leq 10^5$) — количество сегментов пути.

В следующих $l+1$ строках выведите координаты точек ломаной линии в порядке следования (начало пути — конец пути), по два целых числа на каждой строке. Координаты не должны по модулю превосходить 1000. Гарантируется, что есть оптимальный ответ, подходящий под эти условия.

Пример

стандартный ввод	стандартный вывод
3	7
0 0 3 0	3 0
0 1 3 1	6 3
	6 7
	3 10
	0 10
	-3 7
	-3 4
	0 1

Замечание



Задача G. Угадай граф

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Это интерактивная задача

Загадан неориентированный граф из n вершин, соединённых рёбрами. Гарантируется, что исходный граф связен, в нем нет петель и кратных ребер.

Вам необходимо определить, какие рёбра присутствуют в графе. Для этого вы можете взаимодействовать с программой жюри, выполняя запросы о связности графа после удаления некоторых рёбер.

Протокол взаимодействия

В начале работы программа получает одно целое число n ($1 \leq n \leq 40$) — количество вершин в графе.

После этого ваша программа может многократно выводить одну из следующих команд и считывать ответы программы жюри:

- **R i j** — временно удалить ребро между вершинами i и j , если оно существует.
- **U i j** — восстановить ранее удалённое ребро между вершинами i и j (если оно было в исходном графе и вы его удалили операцией R).
- **Q** — запросить у системы, остаётся ли граф связным после текущего набора удалений.

После команды Q программа жюри выведет одно число:

- 1, если граф связен;
- 0, если граф несвязен.

Когда программа определит структуру графа, необходимо вывести результат в следующем формате:

- строка, содержащая единственный символ A;
- далее n строк, каждая из которых содержит n символов 0 или 1, разделенных пробелами.

j -й символ в i -й строке (не считая пробелы) должен быть равен 1, если между вершинами i и j есть ребро, и 0 в противном случае. На главной диагонали всегда должны стоять нули.

После вывода ответа программа должна завершить работу.

Вершины графа нумеруются с единицы. Вы можете сделать не более 900 запросов связности. Каждое ребро можно удалять и восстанавливать многократно. Общее число запросов R и U не может превосходить 10^5 . Загаданный граф фиксирован в начале выполнения программы.

Пример

стандартный ввод	стандартный вывод
4	R 1 2 Q
1	U 1 2 R 3 4 Q
0	R 4 1 Q
0	U 3 4 U 1 4 R 1 2 R 2 3 Q
0	A 0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0

Задача Н. Стоимость задач

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Команда участвует в соревновании по программированию, которое длится t минут. Им дано n задач. Решение можно отправить в любой момент во время соревнования, включая момент ровно через t минут после начала.

Если задача i решена сразу после начала соревнования, команда получает a_i очков. За каждую минуту промедления количество очков за эту задачу уменьшается на b_i . То есть, если задача i решена через x минут после начала, команда получает:

$$a_i - x \cdot b_i$$

Количество очков не может быть отрицательным. На решение задачи i требуется c_i минут.

Необходимо определить максимальное количество очков, которое команда может набрать за время соревнования.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 100\,000$) — длительность соревнования.

Вторая строка содержит одно целое число n ($1 \leq n \leq 500$) — количество задач.

Следующие n строк содержат по три целых числа a_i , b_i , c_i ($1 \leq a_i, b_i, c_i \leq 100\,000$) — количество очков при мгновенном решении, штраф за минуту и время, необходимое для решения i -й задачи.

Формат выходных данных

Выведите одно целое число — максимальное количество очков, которое можно набрать.

Пример

стандартный ввод	стандартный вывод
10	
3	
100 10 5	
200 20 2	
50 5 3	190

Задача I. Биномиальные коэффициенты

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Дано целое число x . Требуется найти такие неотрицательные целые числа n и k , что биномиальный коэффициент

$$C_n^k = \frac{n!}{k!(n-k)!}$$

равен x .

Если существует несколько пар (n, k) , удовлетворяющих этому условию, необходимо выбрать пару с наименьшим значением n . Если таких пар несколько, выбрать из них пару с наименьшим значением k .

Формат входных данных

Ввод содержит одно целое число x ($1 \leq x \leq 10^{100}$).

Формат выходных данных

Выведите два неотрицательных целых числа n и k , такие что $C_n^k = X$, при этом выбранные n и k удовлетворяют описанным выше критериям минимальности.

Примеры

стандартный ввод	стандартный вывод
10	5 2
2025	2025 1

Задача J. Игра со связностью

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Дан неориентированный несвязный граф без петель и кратных ребер из n вершин и m ребер. Демид и Денис играют в игру. За один ход они соединяют ребром две разные вершины графа, между которыми нет ребра. Если после хода игрока граф стал связным, игрок победил. Демид ходит первым. Кто победит при оптимальной игре игроков?

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leq n \leq 150$ $0 \leq m < \frac{n(n-1)}{2}$). Следующие m строк описывают ребра и содержат по два целых различных числа от 1 до n — номера соединяемых вершин. Во входных данных описан изначальный граф.

Формат выходных данных

Выведите Demid, если победит Демид, иначе выведите Denis.

Примеры

стандартный ввод	стандартный вывод
6 5 1 2 3 6 4 5 5 6 6 4	Demid
3 0	Denis

Задача К. Генерация вхождениями шаблонов

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Рассмотрим процесс генерации последовательности нулей и единиц, определяемый набором *шаблонов* — последовательностей символов 0 и 1.

Для получения каждого нового символа выполняются следующие шаги:

1. Рассматриваются все ранее полученные символы.
2. Подсчитывается количество вхождений каждого шаблона в уже полученную последовательность (пересекающиеся вхождения учитываются).
3. Все количества суммируются.
4. Если полученная сумма чётна, следующий символ равен 0, иначе — 1.

Один из шаблонов изначального набора был удалён. По известным оставшимся шаблонам и сгенерированной последовательности определите, сколько различных вариантов возможно для удалённого шаблона.

Удалённый шаблон не может совпадать с уже имеющимися шаблонами и не может быть пустым. Если возможных вариантов бесконечно много, выведите -1 .

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 100\,000$) — количество строк во входных данных.

Следующие n строк содержат непустые строки, состоящие только из символов 0 и 1. Первые $n - 1$ строк задают известные шаблоны, а последняя строка — сгенерированную последовательность.

Суммарная длина всех строк не превышает 10^6 . Все шаблоны различны.

Формат выходных данных

Выполните одно целое число — количество возможных вариантов для удалённого шаблона. Если таких вариантов бесконечно много, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
3 101 0 01101010	2
5 001 1 0 10 01001001	2

Замечание

Пусть заданы шаблоны:

101, 011, 0.

Тогда последовательность будет следующей:

0 1 1 0 1 0 1 0 ...

Пошагово:

1. Первый символ: ни один шаблон не встречается (0 чётно) $\Rightarrow 0$.

2. Второй: одно вхождение шаблона 0 (1 нечётно) $\Rightarrow 1$.
3. Третий: по-прежнему одно вхождение 0 $\Rightarrow 1$.
4. Четвёртый: встречаются 0 и 011 (2, чётно) $\Rightarrow 0$.
5. Пятый: 0 дважды и 011 один раз (3, нечётно) $\Rightarrow 1$.
6. Шестой: 0 дважды, 011 и 101 по одному (4, чётно) $\Rightarrow 0$.
7. Седьмой: 0 трижды, 011 и 101 по одному (5, нечётно) $\Rightarrow 1$.
8. Восьмой: 0 трижды, 011 один раз и 101 дважды (6, чётно) $\Rightarrow 0$.

Задача L. Вкусность конфет

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На столе в ряд выложены n конфет, пронумерованные от 1 до n слева направо. Вкусность i -й конфеты равна a_i .

Необходимо выбрать некоторые конфеты, чтобы их суммарная вкусность была максимальной. При этом выбор ограничен следующим условием:

- среди любых последовательных k конфет можно выбрать не более двух.

То есть, для любого j ($1 \leq j \leq n - k + 1$), среди конфет с номерами от j до $j + k - 1$ можно съесть не более двух.

Требуется определить наибольшую возможную суммарную вкусность выбранных конфет.

Формат входных данных

Первая строка содержит два целых числа n и k ($2 \leq k \leq n \leq 3000$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — вкусности конфет.

Формат выходных данных

Выведите одно число — максимальную суммарную вкусность конфет, которые можно выбрать при соблюдении условия.

Примеры

стандартный ввод	стандартный вывод
5 3 8 3 5 2 6	19
6 4 5 1 4 3 2 7	16

Задача М. Перемещения по ПСП

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дана строка круглых скобок длины k ,

$$s = s_1 s_2 \dots s_k,$$

где каждый символ s_i равен либо "(" либо ")" и строка s является правильной скобочной последовательностью.

Позиции в строке нумеруются от 1 до k слева направо. Для каждой позиции i определены три целочисленные стоимости (в секундах):

$$l_i, \quad r_i, \quad p_i,$$

означающие следующее действие, если курсор в данный момент находится на символе с индексом i :

- переход курсора на один символ влево (в позицию $i - 1$) занимает l_i секунд; если $i = 1$, то такое действие ничего не делает и стоит 0 секунд;
- переход курсора на один символ вправо (в позицию $i + 1$) занимает r_i секунд; если $i = k$, то такое действие ничего не делает и стоит 0 секунд;
- телепортация курсора на позицию, содержащую скобку, соответствующую (matching) текущей скобке в позиции i (то есть на позицию j такую, что скобки на i и j образуют пару в корректном разборе), занимает p_i секунд.

Обозначим через $\text{match}(i)$ соответствующую позицию парной скобки для позиции i (для каждой i такое положение однозначно задано условием корректности строки s).

Дано q запросов. j -й запрос задаётся парой позиций (s_j, e_j) и означает задачу: найти минимальное время n_j (в секундах), необходимое, чтобы при оптимальной последовательности действий переместить курсор с позиции s_j в позицию e_j .

Формат входных данных

Первая строка содержит два целых числа k и q ($2 \leq k \leq 10^5$, $1 \leq q \leq 10^5$) — длину строки и количество запросов.

Вторая строка содержит строку s длины k , состоящую из символов "(" и ")", причём строка s является правильной скобочной последовательностью.

Третья строка содержит k целых чисел l_1, l_2, \dots, l_k ($1 \leq l_i \leq 10^6$).

Четвёртая строка содержит k целых чисел r_1, r_2, \dots, r_k ($1 \leq r_i \leq 10^6$).

Пятая строка содержит k целых чисел p_1, p_2, \dots, p_k ($1 \leq p_i \leq 10^6$).

Шестая строка содержит q целых чисел s_1, s_2, \dots, s_q ($1 \leq s_j \leq k$).

Седьмая строка содержит q целых чисел e_1, e_2, \dots, e_q ($1 \leq e_j \leq k$).

Гарантируется, что суммарные значения k и q не превышают 10^6 .

Формат выходных данных

Выполните q целых чисел в одной строке через пробел — n_j для всех запросов, в том порядке, в котором они даны на вход.

Пример

стандартный ввод	стандартный вывод
12 5 ((((((()))))) 1 7 4 4 12 5 12 11 10 1 6	5 1 2 1 1

Замечание

Переходы влево/вправо и телепортации могут чередоваться произвольным образом. Начальное положение курсора для ответа на каждый запрос равно s_j (то есть запросы независимы друг от друга).

Правильная скобочная последовательность — частный случай скобочной последовательности, определяющийся следующими образами:

- пустая строка есть правильная скобочная последовательность;
- пусть s — правильная скобочная последовательность, тогда (s) правильная скобочная последовательность;
- пусть s_1, s_2 — правильные скобочные последовательности, тогда s_1s_2 правильная скобочная последовательность.

Задача N. Хорошие раскраски – 8

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Дан неориентированный граф без петель и кратных ребер из n вершин и m ребер. Из каждой вершины выходит не более d ребер (степень вершины не превосходит d). Нужно покрасить ребра графа в $d + 1$ цвет так, чтобы никакие два ребра, имеющие общую вершину, не имели один цвет.

Формат входных данных

Первая строка содержит три целых числа n, m и d ($2 \leq n \leq 200$, $1 \leq m \leq 2000$, $1 \leq d \leq n - 1$). Следующие m строк описывают ребра и содержат по два целых различных числа от 1 до n — номера соединяемых вершин.

Формат выходных данных

Если покрасить граф невозможно, выведите -1 . Иначе выведите m целых чисел от 1 до $d + 1$ — цвет очередного ребра (в порядке их перечисления на входе). Если решений несколько, выведите любой. Если можно не использовать какой-то цвет, можно его не использовать.

Пример

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