

Задача **Суп.** Вид из окна

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

На протяжении многих лет Вася работает программистом в одной очень большой и очень известной компании. Эта компания обеспечивает своих сотрудников всем необходимым для приятной и плодотворной работы: бесплатными обедами, транспортом от дома до места работы и многим, многим другим. И вот в один прекрасный солнечный день Вася понял, что ему очень наскучил вид из окна его офиса, и ему нужно, чтобы за окном было что-то новое и прекрасное. А что может быть лучше чудесного горного пейзажа? Придя к этой мысли, Вася попросил своего менеджера подобрать себе новый офис с красивым видом на горы.

В той местности, где располагается офис Васи, каждая гора принадлежит некоторой горной цепи. Так как Васе хочется, чтобы вид из окна его офиса был идеальным, то он попросил подобрать себе такой офис, чтобы никакие две горные цепи, видимые из окна, не пересекались. Менеджер Васи нашел прекрасный новый офис, из которого видно N горных цепей, но он никак не может определить, понравится ли Васе вид из окна этого офиса. Помогите ему!

Более формально, вид из окна офиса представляет собой набор горных цепей, пронумерованных от 1 до N , где горная цепь с номером i представляет собой ломаную на плоскости из l_i звеньев с вершинами в точках $(x_{i,j}, y_{i,j})$, причем для любых i, j выполнено $x_{i,j} < x_{i,j+1}$.

Кроме этого, окно в офисе имеет фиксированную ширину, поэтому все горные цепи начинаются и заканчиваются на одной вертикали, то есть существуют такие числа A и B , что для любого номера i горной цепи выполнено $x_{i,0} = A, x_{i,l_i} = B$.

Отметим, что из определения горной цепи следует, что для любого значения абсциссы $A \leq x \leq B$ на ломаной с номером i существует единственная точка $(x, y_i(x))$ с этим значением абсциссы, принадлежащая этой ломаной. Будем говорить, что горная цепь i находится *строго выше* горной цепи j в точке x , если выполнено строгое неравенство $y_i(x) > y_j(x)$.

Естественно считать, что цепь под номером i *пересекается* с цепью под номером j , если существуют такие два значения абсциссы x_1, x_2 , что цепь i находится строго выше цепи j в точке x_1 , но при этом цепь j находится строго выше цепи i в точке x_2 , то есть выполнены неравенства $y_i(x_1) > y_j(x_1)$ и $y_j(x_2) > y_i(x_2)$. Обратите внимание на поясняющие рисунки, расположенные в примечании к задаче.

Вам необходимо определить, подойдет ли подобранный офис Васе, и, если нет, то найти любую пару *пересекающихся* горных цепей.

Формат входных данных

В первой строке входных данных через пробел идут два целых числа: A и B ($-10^9 \leq A < B \leq 10^9$).

Во второй строке входных данных находится единственное число N — количество горных цепей, видимых из окна выбранного менеджером Васи офиса ($1 \leq N \leq 100\,000$).

Далее следуют описания N горных цепей. В первой строке i -го описания содержится число $l_i \geq 1$ — количество звеньев ломаной, из которых состоит соответствующая горная цепь. В следующих $l_i + 1$ строках описания содержатся два целых числа — координаты $(x_{i,j}, y_{i,j})$ вершин ломаной ($0 \leq j \leq l_i$). Суммарное число звеньев всех ломаных не превосходит 200 000.

Гарантируется, что для каждой горной цепи вершины соответствующей ей ломаной идут во входных данных в порядке возрастания абсциссы, причем для любого i выполнено $x_{i,0} = A, x_{i,l_i} = B$.

Формат выходных данных

Если же офис подходит Васе, то есть никакие две горные цепи из входных данных не пересекаются, в единственной строке выходных данных выведите слово «Yes» (без кавычек).

Иначе выведите в первой строке слово «No» (без кавычек), а на следующей строке выведите два числа — номера двух пересекающихся горных цепей. Горные цепи нумеруются согласно их появлению во входных данных, начиная с 1.

Примеры

ВВОД	ВЫВОД
-3 3 2 1 -3 2 3 1 2 -3 2 0 4 3 2	Yes
0 4 2 3 0 3 1 3 3 1 4 1 1 0 2 4 2	No 1 2

Замечание

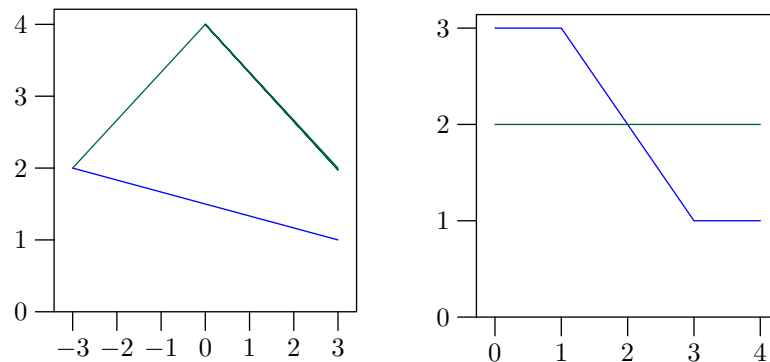


Рис. 1: Пояснения к тестам из условия.

Напоминаем, что *абсциссой* точки называется её x -координата, а *ординатой* — её y -координата.

В первом примере хотя ломаные и касаются друг друга в точке $(-3, 2)$, но, согласно данному выше определению, они **не пересекаются**.

Во втором примере в точке $x_1 = 1$ одна ломаная выше другой, а в точке $x_2 = 3$ — наоборот, то есть горные цепи пересекаются.

Система оценки

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

Группа	Тесты	Баллы	Дополнительные ограничения			Комментарий
			N	l_i	$x_{i,j}, y_{i,j}$	
0	1–2	0	–	–	–	Тесты из условия.
1	3–21	20	$N \leq 50$	$l_i \leq 50$	$ x_{i,j} \leq 50, y_{i,j} \leq 10^6$	
2	22–31	20	$N \leq 100$	$l_i \leq 100$	$ x_{i,j} , y_{i,j} \leq 10^6$	
3	32–41	20	$N \leq 400$	$l_i \leq 400$	$ x_{i,j} , y_{i,j} \leq 10^9$	
4	–	40	$N \leq 100\,000$	$\sum_{i=1}^N l_i \leq 200\,000$	$ x_{i,j} , y_{i,j} \leq 10^9$	Offline-проверка.

Задача Gold. Кладовка из Простоквашино

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Сразу же после заселения в новый дом в Простоквашино кот Матроскин, Шарик и дядя Фёдор затеяли ремонт. Непосредственно перед его началом они обнаружили, что в доме отсутствует кладовка для стройматериалов, и наспех пристроили её к дому. Как только вспомогательное строение было готово, встал вопрос о необходимости провести туда электричество и повесить лампочку.

Обсудив вопросы электрификации новых помещений с почтальоном Печкиным, наши герои узнали много полезной информации. Чтобы посетители не мучились с выбором, в деревенском магазине продаётся только один вид лампочек, зато в неограниченных количествах. Стоит одна лампочка ни дорого, ни дёшево, а ровно C рублей. Правда, лампочки в магазине не самые качественные, и включить каждую из них можно только K раз, а на $K + 1$ включение она перегорает. Недостаток этот компенсируется тем, что во включенном состоянии лампочка перегореть не может. К сожалению, электроэнергия в Простоквашино недешевая, и каждая минута работы лампочки обойдется дяде Фёдору и его друзьям в D рублей.

Узнав всё это, экономный Матроскин составил поминутный график из N предполагаемых посещений кладовки. Каждый визит в новое помещение задаётся моментом входа a_i и моментом выхода b_i . Таким образом, i -й визит продолжается ровно $b_i - a_i$ минут.

Разумеется, во время посещения свет в кладовке должен быть включён. Если в начале очередного визита лампочка не горит, то посетитель сразу её включает, а вот уходя он может как выключить свет, так и оставить его включенным. Если во время очередного включения лампочка перегорает, то её приходится немедленно заменить. Теперь Матроскина интересует минимальное количество рублей, которое придётся потратить, чтобы выполнить все запланированные визиты в кладовку. Изначально в помещении уже висит новая лампочка в выключенном состоянии.

Формат входных данных

В первой строке входных данных записаны четыре целых числа N, K, C, D — количество планируемых посещений кладовки, количество успешных включений для одной лампочки, стоимость покупки лампочки и стоимость минуты работы лампочки соответственно ($1 \leq N, K \leq 200\,000$, $1 \leq C, D \leq 10^9$).

В следующих N строках даны по два целых положительных числа a_i и b_i , описывающих предполагаемые визиты в кладовку ($1 \leq a_i < b_i \leq 10^9$). Посещения не пересекаются по времени и упорядочены, то есть $b_i < a_{i+1}$.

Формат выходных данных

Выведите одно целое число — минимальное количество рублей, которое придётся потратить жителям дома, чтобы выполнить все запланированные визиты в кладовку при свете.

Примеры

ВВОД	ВЫВОД
1 2 5 6 3 5	12
3 1 15 10 1 3 4 5 30 35	105

Замечание

В первом примере достаточно заплатить только за электроэнергию: лампочка должна быть включена на третьей минуте и выключена на пятой, стало быть, суммарные затраты составляют $(5 - 3) \times 6 = 12$.

Во втором примере выгодно не выключать лампочку между первым и вторым посетителем, а для третьего использовать уже новую лампочку.

Система оценки

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

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			N, K	a_i, b_i	
0	1–2	0	–	–	Тесты из условия.
1	3–13	20	$N \leq 50, K = 1$	$a_i, b_i \leq 100$	
2	14–26	20	$N, K \leq 100$	–	
3	27–39	20	$N, K \leq 3\,000$	–	
4	–	40	$N, K \leq 200\,000$	–	Offline-проверка.

Задача **Magenta**. The Cake is a Lie

Ограничение по времени: 5 секунд
Ограничение по памяти: 512 мегабайт

По завершении программы вам предложат торт и возможность выговориться перед дипломированным сочувственным слушателем.

GLaDOS

GLaDOS приготовила для Челл новое, последнее испытание! В его рамках ей предстоит найти торт. Но всё не так-то просто — у неё отобрали порталную пушку и завязали глаза. Единственное, что Челл известно, — что она находится в прямоугольной комнате, выложенной квадратной плиткой, и что где-то в этой комнате расположен торт. Всё что остаётся делать Челл, — слепо переходить с плитки на плитку в поисках торта.

Кроме того, так как действие происходит в Лаборатории Исследования Природы Порталов, на каждой стене комнаты расположен портал, связанный с противоположной стеной. Формально: рассмотрим комнату как прямоугольную сетку размера $W \times H$ и введем систему координат так, чтобы левая нижняя ячейка сетки имела координаты $(0, 0)$, а правая верхняя — $(W - 1, H - 1)$. За один шаг Челл может перейти в одну из четырёх соседних ячеек, причём попытка пройти сквозь стену приводит к тому, что Челл появляется с противоположной стороны комнаты. Например, шаг вниз из ячейки $(x, 0)$ приведёт в ячейку $(x, H - 1)$, а шаг влево из ячейки $(0, y)$ приведёт в ячейку $(W - 1, y)$.

Чтобы сделать испытание сложнее, GLaDOS не сообщила Челл ни размеры комнаты, ни координаты её изначального положения, ни координаты торта. Более того, так как у Челл завязаны глаза, а порталная технология достигла совершенства в своём развитии, Челл даже не может определить, прошла ли она очередным ходом через портал или нет.

Челл может найти торт, только оказавшись в одной клетке с ним. Помогите ей пройти последнее испытание!

Формат взаимодействия с тестирующей системой

Это интерактивная задача.

Ваша программа будет общаться с тестирующей системой по протоколу, описанному ниже. Вам разрешается произвести **не более 200 000 ходов**.

Чтобы переместиться в соседнюю ячейку, выведите строку, содержащую ровно один символ, задающий направление перемещения: «U» — вверх; «D» — вниз; «L» — влево; «R» — вправо.

Затем вы должны считать строку, в которой будет находиться ровно один символ, обозначающий результат перемещения:

- «Y» — после произведённого хода Челл оказалась в клетке с тортом;
- «N» — после произведённого хода Челл оказалась в клетке, не содержащей торт;
- «E» — служебный символ, обозначающий, что после произведённого хода Челл всё ещё не нашла торт, а ваша программа превысила ограничение на количество ходов.

Обратите внимание, после считывания символа «Y» или «E» вы **обязательно** должны сразу завершить вашу программу. В противном случае, вердикт тестирующей системы может быть некорректным!

Гарантируется, что в клетке, в которой исходно находится Челл, нет торта.

В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер вывода — для этого используйте команды `flush(output)` на языке Паскаль или `Delphi`, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Задача Olive. Урок физкультуры

Имя входного файла:	<code>input.txt</code> или стандартный поток ввода
Имя выходного файла:	<code>output.txt</code> или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Феоктист Всеволодович — преподаватель физкультуры старой закалки, глубоко убеждённый, что в начале каждого урока школьников необходимо построить по росту. Для этого он сначала просит школьников построиться самостоятельно, после чего последовательно меняет местами произвольную пару стоящих рядом учеников, пока шеренга не примет желанный вид.

Всего на урок пришло N детей, изначально построившихся таким образом, что рост стоящего на позиции i равен h_i (используется нумерация с 1). Можно считать, что все числа h_i различны и лежат в диапазоне от 1 до N . Шеренга считается упорядоченной, если на первой позиции стоит школьник ростом один, на второй позиции стоит школьник ростом два и так далее.

Феоктист Всеволодович получает большое удовольствие от процесса упорядочивания школьников, поэтому он всегда выбирает наиболее длинную последовательность обменов. С другой стороны, он не хочет чтобы ученики догадались о том, что он умышленно затягивает построение, поэтому никогда не делает заведомо бессмысленных обменов. А именно, преподаватель никогда не меняет местами школьников на позициях i и j , если $h_i < h_j$. Очевидно, что данное ограничение делает процесс сортировки шеренги по росту конечным.

Староста Саша очень любит играть в волейбол и прекрасно понимает, что чем дольше преподаватель будет расставлять всех по местам, тем меньше времени останется для игры. Ученики уже построились некоторым образом, а Феоктист Всеволодович вышел поговорить по телефону, так что Саша может успеть поменять местами ровно двух школьников, необязательно стоящих рядом в шеренге. Разумеется, он хочет сделать это таким образом, чтобы преподаватель как можно быстрее закончил упорядочивать шеренгу (Саша давно уже раскусил, как именно действует Феоктист Всеволодович). С информатикой у старосты всегда были определённые проблемы, поэтому ему требуется ваша помощь.

Формат входных данных

В первой строке ввода содержится единственное число N — количество школьников на уроке ($1 \leq N \leq 1\,000\,000$).

Во второй строке записано N различных целых чисел h_i ($1 \leq h_i \leq N$). i -е число соответствует росту школьника стоящего на i -й позиции.

Формат выходных данных

Выведите два числа — номера позиций школьников, которым необходимо поменяться местами, чтобы минимизировать количество действий преподавателя. Если таких пар несколько, то выведите любую из них. Если никому меняться местами не нужно, выведите `-1 -1`.

Примеры

ВВОД	ВЫВОД
5 2 4 3 5 1	2 5
4 1 2 3 4	-1 -1
10 2 3 7 1 5 10 4 6 9 8	3 7

Замечание

В первом примере из условия после Сашиней перестановки, получится последовательность $\{2, 1, 3, 5, 4\}$, и тренер сможет сделать всего два обмена, перед тем как последовательность станет упорядоченной (например, он может поменять местами 1-го и 2-го школьника, а затем 4-го и 5-го).

Если Саша поменяет местами двух других школьников, тренер затем сможет сделать три или более обменов.

Система оценки

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

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
0	1–3	0	–	Тесты из условия
1	4–9	10	$N \leq 50$	
2	10–15	10	$N \leq 300$	
3	16–21	10	$N \leq 1\,000$	
4	22–27	10	$N \leq 2\,000$	
5	28–33	10	$N \leq 5\,000$	
6	–	10	$N \leq 15\,000$	Offline-проверка.
7	–	10	$N \leq 75\,000$	Offline-проверка.
8	–	10	$N \leq 200\,000$	Offline-проверка.
9	–	10	$N \leq 500\,000$	Offline-проверка.
10	–	10	$N \leq 1\,000\,000$	Offline-проверка.