Задача Satan. Нарисуй ломаные

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

Это интерактивная задача.

Вам даны n точек $A_i = (x_i, y_i)$ на плоскости. Известно, что все x_i различны и все y_i различны.

Ваша задача — рисовать на плоскости ломаные линии, соединяющие эти n точек.

Ломаная линия задается перестановкой $p_1, p_2, \ldots, p_n$ чисел от 1 до n. Ломаная линия состоит из $n-1$ отрезков, первый отрезок соединяет точки A_{p_1} и A_{p_2}, второй отрезок соединяет точки A_{p_2} и A_{p_3}, $\ldots$, последний отрезок соединяет точки $A_{p_{n-1}}$ и A_{p_n}. Обратите внимание, что отрезки могут пересекаться.

Остротой ломаной линии назовем количество индексов $2 \leqslant i \leqslant n-1$, таких что угол $\angle A_{p_{i-1}} A_{p_i} A_{p_{i+1}}$ острый, то есть строго меньше $90°$.

Вам нужно решить четыре задачи:

1. Найти любую ломаную линию, которая имеет максимальную возможную остроту.
2. Дано некоторое целое число c. Найти любую ломаную линию, острота которой $\leqslant c$.
3. Дано некоторое целое число c.

   Ответить на q запросов, каждый задается единственным целым числом k_i ($c \leqslant k_i \leqslant n-c$). В i-м запросе нужно построить ломаную линию, которая имеет остроту ровно k_i.
4. Дано некоторое целое число c.

   Для каждого k от c до $n-c$ построить ломаную $p^{(k)}$, имеющую остроту ровно k. В качестве ответа предоставить $n-2c+1$ чисел $\mathrm{hash}\left(p^{(c)}\right), \mathrm{hash}\left(p^{(c+1)}\right), \ldots, \mathrm{hash}\left(p^{(n-c)}\right)$, где $hash(p) = \left(\sum\limits_{i=1}^{n} p_i b^{i-1}\right) \bmod m$ — полиномиальный хеш перестановки p с параметрами $b = 10^6 + 3$ и $m = 10^9 + 7$.

   Затем нужно ответить на q запросов, каждый задается единственным целым числом k_i ($c \leqslant k_i \leqslant n-c$). В i-м запросе нужно предоставить ломаную $p^{(k_i)}$. Будет проверено, что острота этой ломаной ровно k_i и ее хеш равен предоставленному ранее значению $\mathrm{hash}\left(p^{(k_i)}\right)$.

   Обратите внимание, что запросы будут появляться после получения хешей.

Протокол взаимодействия

В первой строке находятся два целых числа task, group ($1 \leqslant \text{task} \leqslant 4$, $0 \leqslant \text{group} \leqslant 21$) — номер задачи, которую надо решить в этом тесте и номер группы теста.

Во второй строке находится единственное целое число n ($3 \leqslant n \leqslant 80\,000$) — количество точек на плоскости.

В каждой из следующих n строк находятся два целых числа x_i, y_i ($|x_i|, |y_i| \leqslant 10^9$) — координаты точек. Гарантируется, что все x_i различны и все y_i различны.

Если $\text{task} = 1$, то на этом входные данные заканчиваются и вы должны вывести любую перестановку, острота которой максимально возможная. На этом взаимодействие завершается.

Если $\text{task} \neq 1$, то в следующей строке находится единственное целое число c ($2 \leqslant c \leqslant \frac{n}{2}$).

Если $\text{task} = 2$, то на этом входные данные заканчиваются и вы должны вывести любую перестановку, острота которой $\leqslant c$. На этом взаимодействие завершается.

Если $\text{task} = 4$, то ваше решение должно вывести $n-2c+1$ целых чисел $\mathrm{hash}\left(p^{(c)}\right), \mathrm{hash}\left(p^{(c+1)}\right), \ldots, \mathrm{hash}\left(p^{(n-c)}\right)$, где $0 \leqslant \mathrm{hash}\left(p^{(i)}\right) < 10^9 + 7$. Обратите внимание, что этого не надо делать, если $\text{task} = 3$.

Дальнейшее взаимодействие происходит только при $\text{task} = 3$ или $\text{task} = 4$.

В следующей строке находится единственное целое число q ($1 \leqslant q \leqslant 50$) — количество запросов.

Затем q раз в очередной строке появляется запрос k_i ($c \leqslant k_i \leqslant n - c$). В качестве ответа вы должны в отдельной строке вывести перестановку. Острота этой перестановки должна быть ровно k_i. Если $\text{task} = 4$, то хеш этой перестановки должен совпадать с предоставленным ранее хешом.

Поскольку задача интерактивная, после вывода каждой строки не забывайте выводить символ перевода строки и сбрасывать буфер выходного потока.

Система оценки

Тесты к этой задаче состоят из двадцать одной группы. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых для нее групп.

Группа	Баллы	Ограничения				Необх. Группы	Комментарий
		task	n	c	Доп. ограничения		
0	0	–	–	–	–	–	Тесты из условия.
1	8	1	$n \leqslant 20\,000$	–	$x_i < x_{i+1}$, $y_i < y_{i+1}$	–	
2	6	1	$n \leqslant 10$	–	точки случайные	–	
3	5	1	$n \leqslant 1000$	–	точки случайные	2	
4	5	1	$n \leqslant 20\,000$	–	точки случайные	2 – 3	
5	6	1	$n \leqslant 20\,000$	–	–	1 – 4	
6	17	2	$n = 80\,000$	$c = 800$	–	–	
7	7	3	$n = 80\,000$	$c = 800$	$x_i < x_{i+1}$, $y_i < y_{i+1}$	–	
8	4	3	$n = 50$	$c = 25$	точки случайные	–	
9	4	3	$n = 200$	$c = 80$	точки случайные	–	
10	4	3	$n = 1000$	$c = 300$	точки случайные	–	
11	3	3	$n = 5000$	$c = 600$	точки случайные	–	
12	3	3	$n = 80\,000$	$c = 35\,000$	точки случайные	–	
13	3	3	$n = 80\,000$	$c = 5000$	точки случайные	12	
14	3	3	$n = 80\,000$	$c = 2000$	–	12 – 13	
15	2	3	$n = 80\,000$	$c = 800$	–	7, 12 – 14	
16	6	4	$n = 80\,000$	$c = 800$	$x_i < x_{i+1}$, $y_i < y_{i+1}$	–	
17	3	4	$n = 5000$	$c = 600$	точки случайные	–	
18	3	4	$n = 80\,000$	$c = 35\,000$	точки случайные	–	
19	3	4	$n = 80\,000$	$c = 5000$	точки случайные	18	
20	3	4	$n = 80\,000$	$c = 2000$	–	18 – 19	
21	2	4	$n = 80\,000$	$c = 800$	–	16, 18 – 20	

В тех группах, в которых указано, что точки случайные, все координаты всех точек x_i, y_i сгенерированы случайно равновероятно на отрезке $[-10^9, 10^9]$.

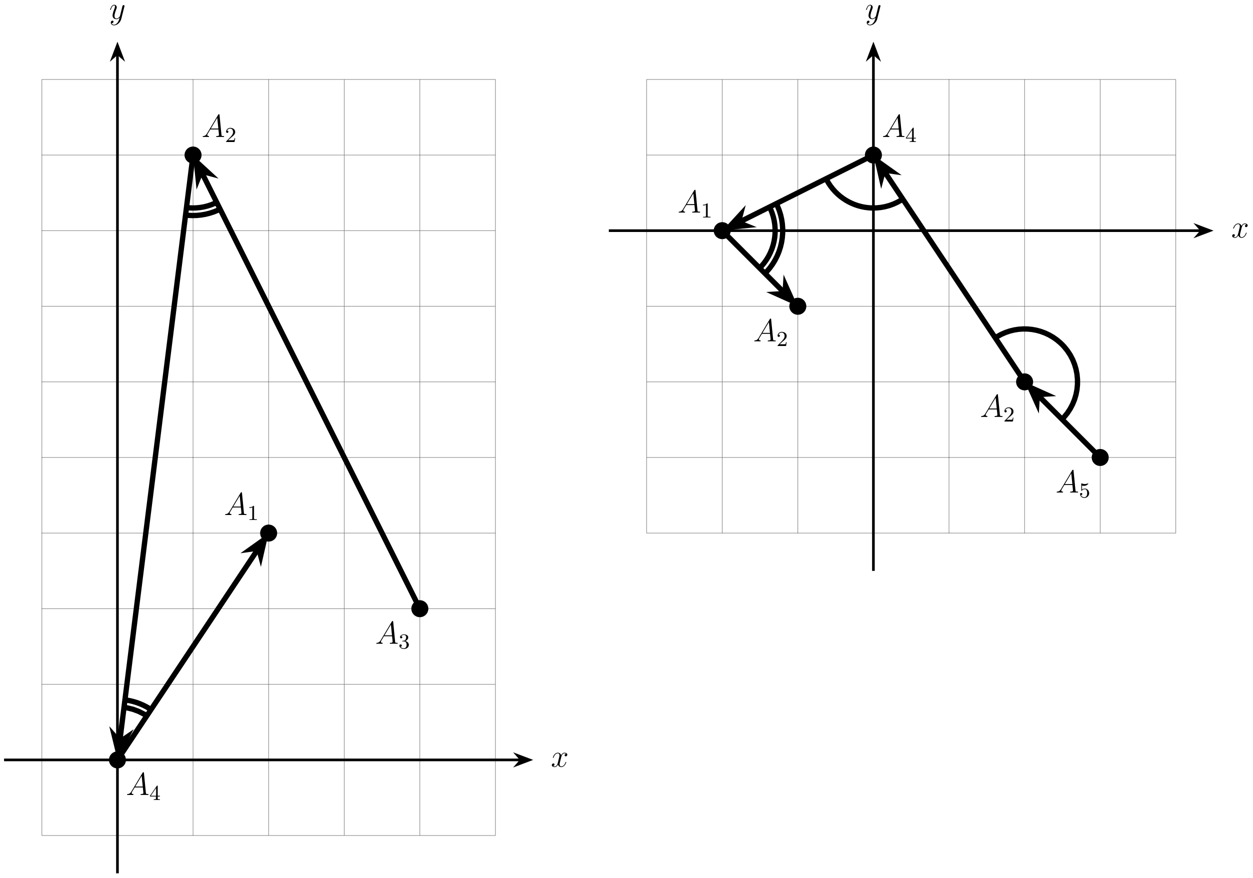
Примеры

ввод	вывод
1 0 4 2 3 1 8 4 2 0 0	3 2 4 1
2 0 5 -2 0 -1 -1 0 1 2 -2 3 -3 2	5 4 3 1 2
3 0 6 0 0 1 1 2 2 3 -3 4 -2 5 -1 2 3 2 3 4	1 2 3 4 5 6 4 5 6 1 3 2 6 2 4 3 5 1
4 0 5 -2 -1 -1 1 1 6 0 -3 2 0 2 2 2 3	534735187 776162084 4 5 1 2 3 1 3 2 5 4

Пояснение

На всех картинках двумя дугами обозначены острые углы, одной дугой обозначены не острые углы.

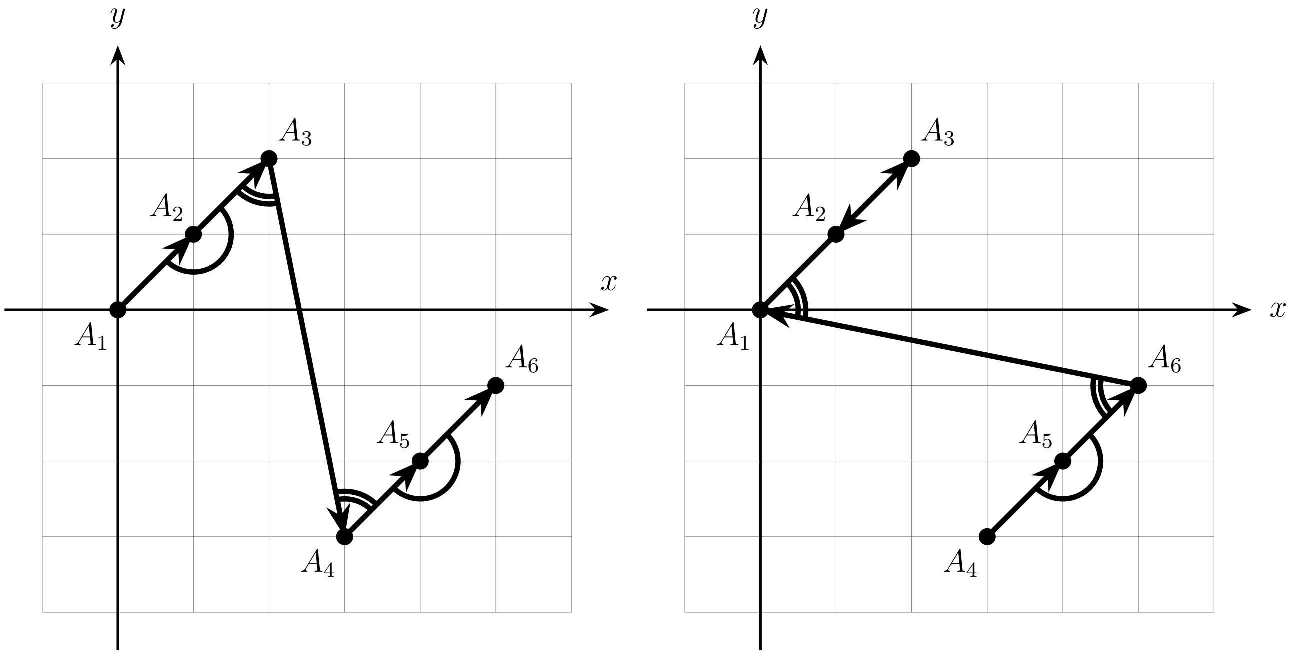
Первый и второй тесты из условия

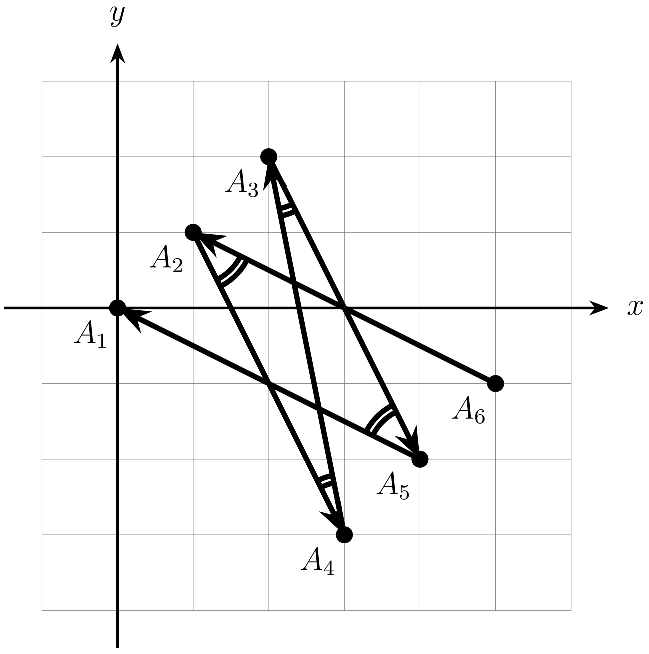


В первом тесте из условия оба угла ломаной острые, поэтому ломаная имеет максимально возможную остроту 2.

Во втором тесте из условия острота равна 1, она $\leqslant 2$, поэтому ломаная подходит.

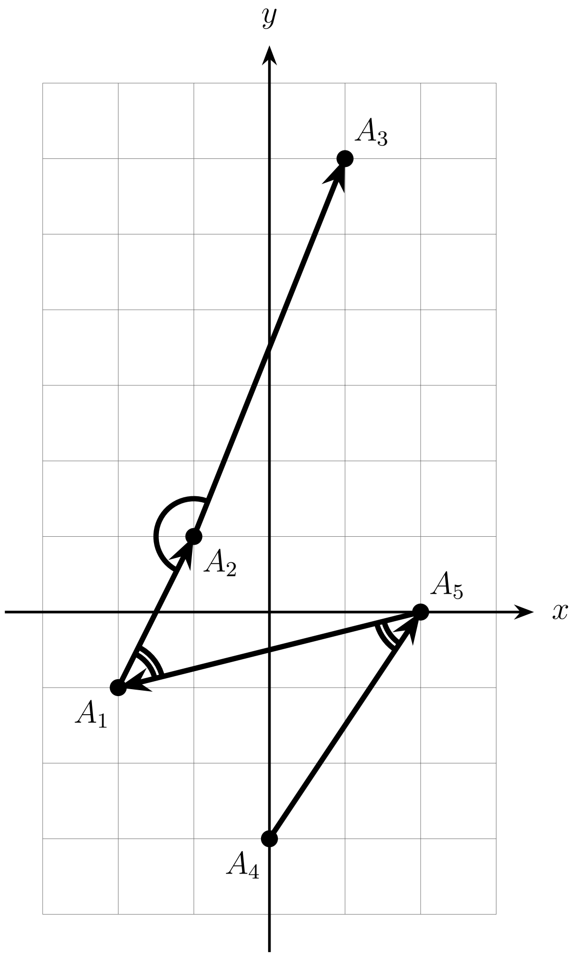
Третий тест из условия

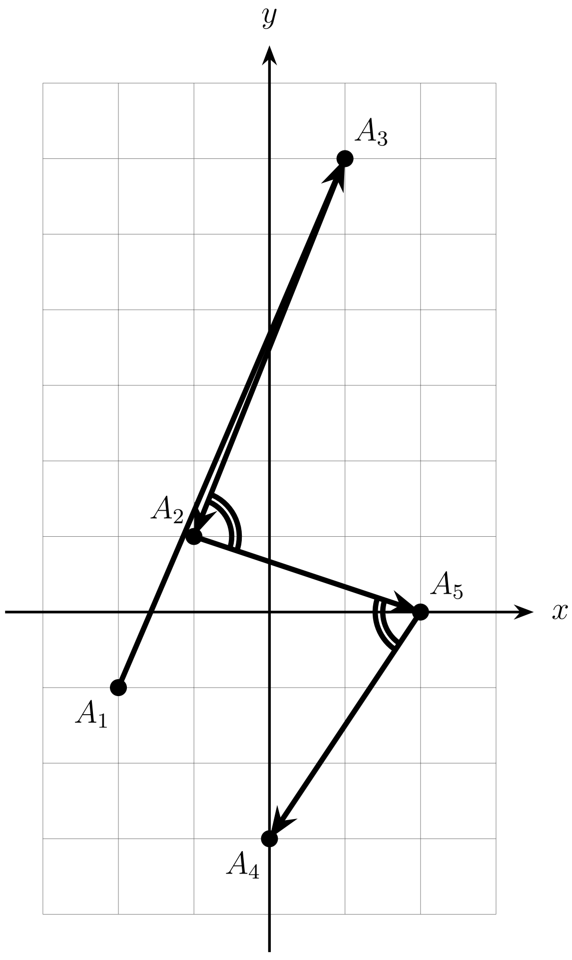




В третьем тесте из условия мы строим ломаные, которые имеют остроту 2, 3, 4.

Четвертый тест из условия





В четвертом тесте из условия мы последовательно строим ломаные, которые имеют остроту 2, 3. При этом заранее были предоставлены хеши ломаных, которые совпадают с хешами приведенных далее ломаных.

Задача Sherlock Holmes. Доска улик

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

Володя мечтает стать детективом. Поэтому Володя нередко читает книги, в которых рассказывается об невероятных историях раскрытия преступных группировок. Изучая очередное дело Володя наткнулся на интересные подробности следствия.

Всего в деле было n подозреваемых. На доске улик были изображены все n подозреваемых. Изначально у детективов не было никаких связей между подозреваемыми.

Однако в ходе расследования поочерёдно возникали новые улики. Каждая улика связывала двух подозреваемых, причём ранее эти подозреваемые не имели никакой связи друг с другом, даже косвенной, через несколько других подозреваемых.

Рассмотрим, что происходило, когда у детективов возникала улика, указывающая на связь между подозреваемыми A и B. Кроме имён подозреваемых, у каждой улики было три параметра: c_A — сила доказательства против A, c_B — сила доказательства против B, а также w_{AB} — общая сила улики. По естественным соображениям сила улики не могла превышать суммарную силу доказательств против A и B, то есть для каждой улики **обязательно** выполнялось $w_{AB} \leqslant c_A + c_B$. Получив такую улику, детективы проводили на доске ребро (линию) между изображениями A и B, назначая этому ребру вес равный силе улики, w_{AB}. А также на изображение подозреваемого A наклеивался стикер с числом c_A, а на B наклеивался стикер с числом c_B. Причём, если на изображении уже были другие стикеры, новый стикер наклеивался поверх старых.

Дело было раскрыто ровно в тот момент, когда все подозреваемые оказались связаны, через $n-1$ улику. После раскрытия преступления, доска в неизменном виде была помещена в музей.

Вдохновлённый таким подходом Володя посетил музей, в котором сохранилась доска с этого расследования, и подробно изучил ее. Володя увидел, что изображение подозреваемого v содержало стикеры с числами $c_{v,1}, \ldots, c_{v,deg_v}$ пронумерованных **от верхнего к нижнему**. Здесь deg_v обозначает количество улик, связанных с подозреваемым v. Также Володя запомнил, что i-я улика соединяла подозреваемых a_i и b_i и имела силу доказательства w_i, однако улики были пронумерованы произвольным образом и их номера не обязательно соответствовали тому порядку, в котором они появлялись во время следствия.

Из-за путаницы с номерами улик, информация на доске не помогала воссоздать полную картину следствия. Чтобы полностью восстановить историю дела, Володе необходимо восстановить любой возможный хронологический порядок, в котором улики могли появляться у детективов. Но эта задача для него непосильно трудна. Помогите ему! Если есть несколько возможных вариантов восстановления, подойдёт любой из них. Также возможна ситуация, что музей подделал часть информации, и подходящего порядка не существует.

Формат входных данных

В первой строке входных данных даны два целых числа n и g ($2 \leqslant n \leqslant 200\,000$, $0 \leqslant g \leqslant 9$) — количество подозреваемых в деле и номер группы тестов.

В следующих $n-1$ строках описываются улики. В i-й строке даны три целых числа a_i, b_i и w_i ($1 \leqslant a_i, b_i \leqslant n$, $1 \leqslant w_i \leqslant 10^9$, $a_i \neq b_i$) — номера подозреваемых, которых связывает i-я улика, и общая сила i-й улики. Гарантируется, что улики связывают всех подозреваемых между собой.

В следующих n строках описываются числа, написанные на стикерах. В i-й строке дано deg_i целых чисел $c_{i,1}, \ldots, c_{i,deg_i}$ ($0 \leqslant c_{i,j} \leqslant 10^9$) — числа написанные на стикерах на изображении i-го подозреваемого от верхнего к нижнему. Напомним, что deg_i равняется количеству улик, связанных с подозреваемым i.

Формат выходных данных

Если подходящего под условия задачи хронологического порядка восстановления улик не существует, в единственной строке выведите «`No`» (без кавычек).

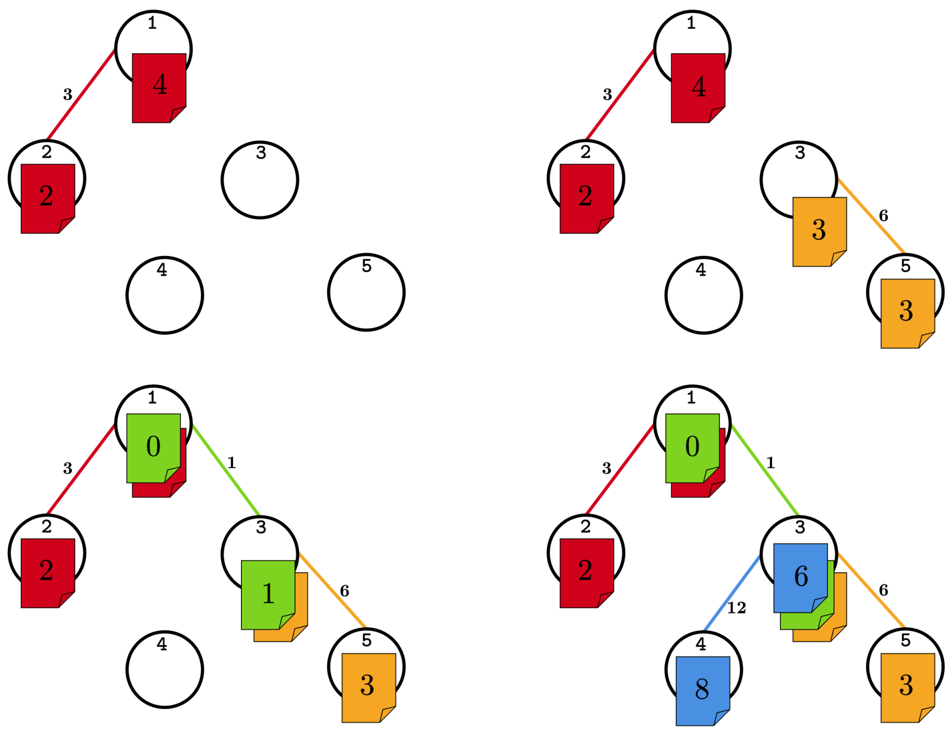
Иначе, в первой строке выведите «Yes» (без кавычек). Во второй строке выведите $n-1$ чисел — подходящий хронологический порядок возникновения улик. Улики пронумерованы от 1 до $n-1$ в таком же порядке, как они заданы во входных данных. Если возможных порядков несколько, выведите любой из них.

Примеры

<table>
<tr><th>ввод</th><th>вывод</th></tr>
<tr><td>5 0
1 2 3
1 3 1
3 4 12
3 5 6
0 4
2
6 1 3
8
3</td><td>Yes
1 4 2 3</td></tr>
<tr><td>7 0
1 2 4
2 3 4
3 4 4
4 5 4
5 6 4
6 7 4
2
1 2
2 3
1 2
3 2
1 2
179</td><td>Yes
5 1 2 3 6 4</td></tr>
<tr><td>4 0
1 2 7
1 3 6
1 4 5
3 2 1
5
4
3</td><td>No</td></tr>
</table>

Пояснение

В первом тесте из условия один из возможных порядков — $[1, 4, 2, 3]$. Первая, в хронологическом порядке, улика связывает $A = 1$ и $B = 2$, $c_A = 4, c_B = 2, w_{AB} = 3$, $3 \leqslant 2 + 4$ — улика корректная. Вторая, в хронологическим порядке, улика связывает $A = 3$ и $B = 5$, $c_A = 3, c_B = 3, w_{AB} = 6$, $6 \leqslant 3 + 3$ — улика корректная. Третья, в хронологическом порядке, улика связывает $A = 1$ и $B = 3$, $c_A = 0, c_B = 1, w_{AB} = 1$, $1 \leqslant 0 + 1$ — улика корректная. Четвёртая, в хронологическом порядке, улика связывает $A = 3$ и $B = 4$, $c_A = 6, c_B = 8, w_{AB} = 12$, $12 \leqslant 6 + 8$ – улика корректная. Для лучшего понимания смотрите иллюстрацию.



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	Доп. ограничения		Необх. Группы	Комментарий
		n	a_i, b_i, c_i, w_i		
0	0	–	–	–	Тесты из условия.
1	10	$n \leqslant 10$	–	0	–
2	15	–	$a_i = i, b_i = i + 1$ для всех i	–	–
3	8	–	$a_i = 1, b_i = i + 1$ для всех i	–	–
4	9	–	$a_i \leqslant 2, b_i = i + 1$ для всех i	3	–
5	7	$n \leqslant 1000$	$c_{i,1} \leqslant c_{i,2} \leqslant \ldots \leqslant c_{i,deg_i}$ для всех i	–	–
6	7	$n \leqslant 1000$	$c_{i,j} = 0$ для всех $1 \leqslant i \leqslant n$ и $j \geqslant 2$	–	–
7	17	–	$\sum\limits_{v=1}^{n} \sum\limits_{i=1}^{deg_v} c_{v,i} = \sum\limits_{i=1}^{n-1} w_i$	–	–
8	16	$n \leqslant 1000$	–	0, 1, 5, 6	–
9	11	–	–	0 – 8	**Offline-проверка.**

Задача Ded Moroz. Больше подарков хороших и разных

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Организаторы Закрытой олимпиады школьников по программированию решили подготовить подарки для участников олимпиады. Всего было заказано k одинаковых коробок с подарками, каждая коробка содержит стопку из n подарков. Вверху каждой стопки лежит подарок типа a_1, под ним подарок типа a_2, и так далее, внизу стопки лежит подарок типа a_n.

Раздача подарков будет происходить следующим образом: сначала будут отдаваться подарки из первой стопки сверху вниз. После того, как в первой стопке не останется подарков, будут отдаваться подарки из второй стопки сверху вниз, и так далее, в конце будут отдаваться подарки из k-й стопки.

Участнику можно раздавать сразу несколько подарков, поэтому в начале подарки будут отдаваться первому участнику, потом второму, и так далее. Известно, что если участнику достанется больше, чем t различных типов подарков, то участник будет слишком радоваться, и плохо напишет олимпиаду. Чтобы участники написали олимпиаду хорошо, было решено выдавать каждому участнику не более t различных типов подарков (при этом участнику может достаться несколько подарков одинакового типа).

Организаторы Закрытой олимпиады хотят сделать олимпиаду максимально эксклюзивной, и пригласить туда как можно меньше участников. Помогите организаторам узнать, какое минимальное число участников они могут позвать, чтобы все подарки были розданы участникам, а каждый получил не более t различных типов подарков.

Формат входных данных

Первая строка входных данных содержит три целых числа n, k и t ($1 \leqslant n \leqslant 300\,000$, $1 \leqslant k, t \leqslant 10^9$) — количество подарков в одной стопке, количество стопок с подарками и максимальное количество различных типов подарков, которое может достаться одному участнику.

Следующая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — типы подарков, в порядке следования сверху вниз стопки.

Формат выходных данных

Выведите единственное число — минимальное число участников, такое что все подарки будут им розданы, а каждый участник получит не более t различных типов подарков.

Примеры

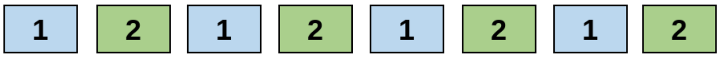
ввод	вывод
`2 4 1` `1 2`	`8`
`4 3 1` `1 1 2 1`	`7`
`7 2 3` `1 2 3 4 5 6 7`	`5`

Пояснение

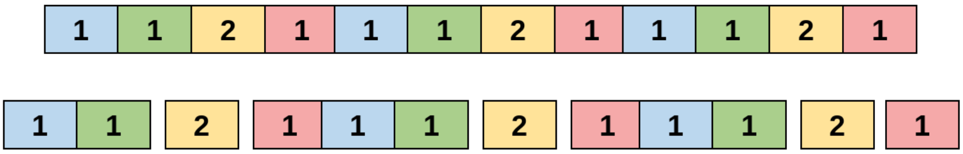
В первом примере стопка содержит следующие типы подарков (в порядке сверху вниз). Различными цветами обозначаются различные позиции в стопке.

Всего есть 4 стопки подарков, соответственно подарки будут отдаваться в следующем порядке:

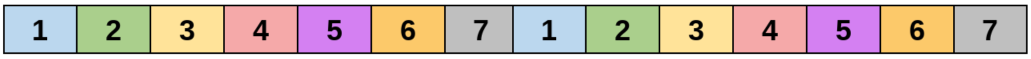
Так как $t = 1$ каждый участник в данном случае может получить только подарки одного типа:



Во втором примере порядок выдачи подарков и итоговые комплекты подарков выглядят следующим образом:



В третьем примере порядок выдачи подарков следующий:



В данном случае одним из возможных оптимальных разбиений подарков на комплекты является следующее разбиение:

Система оценки

Тесты к этой задаче состоят из шести 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.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		n	k	t		
0	0	–	–	–	–	Тесты из условия.
1	14	$n \leqslant 100$	$k \leqslant 10$	–	0	–
2	12	–	–	$t = 1$	–	–
3	16	$n \leqslant 1000$	$k \leqslant 1000$	–	0, 1	–
4	21	$n \leqslant 1500$	$k \leqslant 10^6$	–	0, 1, 3	–
5	18	–	$k \leqslant 10^6$	–	0, 1, 3, 4	–
6	19	–	–	–	0 – 5	–

Задача Scheherazade. Большая хурма

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:	1024 мегабайта

Алиса и Боб купили большую хурму, разрезали её на n кусочков, размерами $w_1, \ldots, w_n$, и сразу начали её есть. Ребята будут есть кусочки одновременно, для каждого из них процесс поедания устроен следующим образом:

Как только кто-то закончил есть свой предыдущий кусочек (а также в самом начале трапезы), он выбирает очередной кусочек и начинает его есть. Если взять кусочек размером w, то на его поедание уйдет ровно w секунд, а затем настанет время выбирать новый кусочек. Если оба одновременно закончили есть свой предыдущий кусочек (или если поедание только началось), то первой кусочек выбирает Алиса, но есть они начнут одновременно. Выбор нового кусочка не занимает времени.

Так как Алиса и Боб оба перфекционисты, когда они выбирают кусочек, то из всех оставшихся кусочков они возьмут либо самый маленький (с наименьшим w_i), либо самый большой (с наибольшим w_i).

Процесс поедания заканчивается, когда последний человек доел свой кусочек.

Алиса и Боб оба заинтересованы в том, чтобы съесть как можно больше хурмы. Найдите суммарный размер кусочков, которые съест Алиса, и которые съест Боб, если они оба будут выбирать кусочки оптимально.

Формат входных данных

В первой строке дано одно целое число n $(1 \leqslant n \leqslant 2000)$ — количество кусочков хурмы.

Во второй строке даны n целых чисел $w_1,\ w_2,\ \ldots,\ w_n$ $(1 \leqslant w_i \leqslant 20\,000,\ w_i \leqslant w_{i+1})$ — размеры кусочков хурмы.

Обозначим за W сумму размеров всех кусочков. Гарантируется, что $W \leqslant 20\,000$.

Формат выходных данных

В одной строке выведите два числа — суммарный размер всех кусочков, которые съест Алиса, и суммарный размер всех кусочков, которые съест Боб, если оба будут выбирать кусочки оптимально.

Примеры

ввод	вывод
5 1 1 3 4 6	8 7
4 1 1 2 2	3 3
4 1 7 7 9	10 14

Пояснение

В первом примере Алисе стоит первым делом взять кусочек, размером 1. Сразу после этого Бобу тоже стоит взять кусочек размером 1. Спустя секунду, Алиса возмет кусочек размером 3, затем Боб возьмет кусочек размером 6. Ещё через 3 секунды Алиса возьмет кусочек размера 4. Ещё через 3 секунды Боб закончит есть, а ещё через секунду поедание закончится. При этом Алиса съест кусочки размерами $1 + 3 + 4 = 8$, а Боб: $1 + 6 = 7$

В третьем примере Алисе стоит взять кусочек размером 1, а Бобу — размером 7. Через секунду Алиса возьмет себе кусочек размером 9, и ещё через 6 секунд, Боб возьмет себе кусочек, размером 9.

Система оценки

Тесты к этой задаче состоят из четырёх 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	Доп. ограничения		Необх. Группы	Комментарий
		n	w_i		
0	0	–	–	–	Тесты из условия.
1	10	$n = 3$	–	–	–
2	12	–	$w_i \leqslant 2$	–	–
3	19	$n \leqslant 200$	$w_i \leqslant 500$	0	–
4	15	$n \leqslant 500$	$W \leqslant 5000$	–	$w_{i+1} \leqslant 2 \cdot w_i$ для всех $1 \leqslant i \leqslant n-1$
5	13	–	–	2, 4	$w_{i+1} \leqslant 2 \cdot w_i$ для всех $1 \leqslant i \leqslant n-1$
6	31	–	–	0 – 5	**Offline-проверка.**