

Задача Е. Перестроение

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

Каждое утро капитан Ё проводит занятия по строевой подготовке в возглавляемой им роте солдат. Всего в роте N солдат, каждый из которых носит форму определенного цвета. Различных цветов формы не более 26, так что для удобства солдаты обозначают цвета строчными латинскими буквами. Таким образом, каждому из N солдат соответствует буква от 'a' до 'z' — цвет его формы.

За многие месяцы службы солдаты выяснили, что капитан пребывает в наилучшем расположении духа в том случае, когда цвета формы солдат в шеренге образуют определенную последовательность. Недолго думая, они выписали соответствующую строку S из N букв на асфальте и договорились, что отныне каждый должен при построении вставлять именно на ту букву, которая соответствует цвету его формы.

Но к 23 февраля солдаты решили удивить капитана и поменяться местами так, чтобы *каждый* солдат встал *не* на ту букву, которая соответствует цвету его формы. Так, солдат с цветом формы 'q' может встать на любую букву, кроме буквы 'q', иначе удивление капитана будет недостаточным.

Помогите солдатам организовать праздничное построение: по данной строке S , обозначающей старую последовательность цветов, выведите строку T , являющуюся перестановкой символов строки S и обозначающую новую последовательность цветов. i -й символ строки T должен отличаться от i -го символа строки S .

Формат входного файла

В первой строке входного файла содержится единственное целое число N — количество солдат в роте ($1 \leq N \leq 100\,000$). Во второй строке содержится строка S , состоящая из N строчных латинских букв.

Формат выходного файла

Единственная строка выходного файла должна содержать искомую строку T , если задумка солдат осуществима, и «Impossible» в противном случае. Если верных ответов несколько, выведите любой из них.

Примеры

<code>shuffling.in</code>	<code>shuffling.out</code>
9 olimpiada	aiadiplom
7 baaaaaa	Impossible

Система оценки

Тесты к этой задаче состоят из четырех групп. Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

0. Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
1. Тесты 3–21. В тестах этой группы $N \leq 9$. Эта группа оценивается в 30 баллов.
2. Тесты 22–36. В тестах этой группы $N \leq 200$, а строка не может содержать никаких букв, кроме 'a', 'b' и 'c'. Эта группа оценивается в 30 баллов независимо от первой.
3. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура, причем только в случае прохождения всех тестов первой и второй групп.

Задача F. Берляндиолимпстрой

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

Скоро в Берляндии пройдет очередная Олимпиада. В рамках подготовки к этому важному мероприятию Берляндиолимпстрой уже возвел N объектов и теперь хочет разобраться с тем, во сколько Берляндии это обошлось.

Стройка длилась $K + 1$ день со дня номер 0 по день номер K , причем стоимость j -го объекта в нулевой день была равна a_j бурлям. Однако каждый следующий день стоимость каждого объекта увеличивалась согласно следующему правилу: стоимость j -го объекта в i -й день становилась равна сумме стоимостей всех объектов с номерами, меньшими или равными j , в предыдущий день. Иначе говоря, $S_{i,j} = \sum_{m=1}^j S_{i-1,m}$, где $S_{i,j}$ — стоимость j -го объекта в i -й день. В итоге на j -й объект было потрачено $S_{K,j}$, то есть его стоимость в последний K -й день. Назовем эту величину *итоговой стоимостью j -го объекта*.

Такие увеличения стоимостей проектов для Берляндии не редкость, однако оказалось, что и этих денег не хватило! Выяснилось, что в некоторый день $i > 0$ стоимость некоторого объекта j дополнительно повысилась на пока не известную следователям сумму X (то есть $S_{i,j} = \sum_{m=1}^j S_{i-1,m} + X$), что повлияло на стоимости объектов в последующие дни. Следователи выяснили, что из-за этого сумма итоговых стоимостей всех объектов увеличилась на R бурлей.

Помогите следователям выяснить минимально возможное значение X .

Формат входного файла

В первой строке входного файла содержатся три целых числа N , K , R : количество олимпийских объектов ($1 \leq N \leq 10^5$), количество дней увеличения стоимости объектов ($1 \leq K \leq 10^5$) и количество бурлей, на которое незаконно возросла итоговая сумма ($1 \leq R \leq 10^{18}$). В следующей строке входного файла содержатся N целых чисел a_i — стоимости объектов в нулевой день ($1 \leq a_i \leq 10^9$).

Формат выходного файла

Единственная строка выходного файла должна содержать единственное целое число — минимально возможное значение X .

Пример

jobbery.in	jobbery.out
3 3 12 1 3 3	2

Система оценки

Тесты к этой задаче состоят из четырех групп.

- Тест 1. Тест из условия, оцениваемый в ноль баллов.
- Тесты 2–25. В тестах этой группы $N \leq 10$, $K \leq 10$, $a_i \leq 10$, искомое значение X не превосходит 10. Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- Тесты 26–38. В тестах этой группы $N \leq 1\,000$, $K \leq 1\,000$. Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы. Решение будет тестироваться на тестах этой группы только в случае прохождения всех тестов первой группы.
- В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура, причем только в случае прохождения всех тестов первой и второй групп. Тесты в этой группе оцениваются **независимо**.

Задача G. Транспортные потоки

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

В родном городе Иннокентия активно ведется строительство новых дорог. В частности, в городе даже появился первый автобусный маршрут и несколько автобусов, которые по этому маршруту курсируют. Маршрут представляет собой N остановок, автобус перемещается между ними последовательно: сначала из автобусного парка на первую остановку, потом с первой остановки — на вторую и так далее. Для каждой пары соседних остановок известно минимальное время, за которое автобус может преодолеть расстояние между ними.

Как известно, автобус должен ходить по расписанию. И Иннокентий, используя свои многочисленные связи в магазине плитки, совершил невозможное: по маршруту теперь курсируют целых M автобусов, и на каждой остановке висит свое расписание, которое представляет собой набор из M времен. Плиточный магнат является крупным авторитетом в городе, поэтому расписание соблюдается: от каждой остановки ровно в каждое

из указанных времен отправляется автобус. Казалось, что проблема общественного транспорта навсегда решена...

Однако, дьявол кроется в деталях. Действительно, автобусы отправляются с остановок в нужные времена, но никто не гарантирует, что между остановками не произойдет обгон, и автобус, который отправился от предыдущей остановки раньше, не отправится со следующей гораздо позже, при этом не нарушая условия, что в каждое из указанных в расписании времен какой-то автобус отправляется.

Иннокентий решил оценить масштабы трагедии. Для этого он попросил каждого из Q своих друзей сообщить маршрут, по которому они добираются до места работы. Каждый маршрут описывается тремя числами u_i , v_i , w_i : u_i — это номер остановки, ближайшей к дому i -го друга, v_i — номер остановки, ближайшей к его работе, а w_i — номер автобуса, на котором i -й друг едет из дома на работу. При этом с точки зрения i -го друга автобусы нумеруются от 1 до M в том порядке, в котором они отправляются с остановки u_i .

Иннокентий просит вас независимо для каждого друга определить, насколько поздно тот может доехать до конечной остановки своего маршрута.

Формат входного файла

В первой строке входных данных содержатся два целых числа N и M — количество остановок и количество автобусов соответственно ($2 \leq N \times M \leq 150\,000$). В следующей строке содержатся $N - 1$ целых чисел $travel_1, \dots, travel_{N-1}$, где $travel_i$ — минимальное время, необходимое для перемещения между остановками i и $i + 1$ ($1 \leq travel_i \leq 10^9$).

В следующих N строках содержатся описания расписаний, каждое из которых представляет собой отсортированный по возрастанию список из M различных целых чисел t_i — времен, в которые автобусы должны отправляться с соответствующей остановки ($1 \leq t_i \leq 10^9$).

В следующей строке содержится число T — тип теста (1 или 2). Если $T = 1$, то это — *обычный тест*. Тогда на следующей строке содержится целое число Q — количество опрошенных друзей Иннокентия ($1 \leq Q \leq 150\,000$). Далее в Q строках содержатся описания маршрутов друзей, каждое из которых состоит из трех целых чисел u_i , v_i и w_i : номеров остановок, где начинается и заканчивается поездка i -го друга, и номер автобуса в расписании остановки u_i , на котором эта поездка совершается ($1 \leq u_i < v_i \leq N$, $1 \leq w_i \leq M$).

Обратите внимание: дальнейшее описание относится только к **последней** группе тестов. Если $T = 2$, то это — *тест-серия*. Тогда на следующей строке содержатся три целых числа — A , B и K ($1 \leq A, B \leq 10^3$, $1 \leq K \leq 150$).

В *тесте-серии* у Иннокентия $Q = (N - 1) \cdot M \cdot K$ друзей. На каждой из $N - 1$ остановок, кроме последней, проживает ровно $M \cdot K$ друзей, причем для каждого w от 1 до M есть ровно K друзей, которые уезжают с этой остановки w -м автобусом.

Остановки, до которых едут K друзей, уезжающих с u -й остановки w -м автобусом, определяются следующим образом. Задается последовательность чисел q_i : $q_1 = A$, $q_2 = B$, для $i > 2$ $q_i = u \cdot q_{i-1} + w \cdot q_{i-2} + 42$. Тогда i -й из этих K друзей будет ехать до остановки с номером $v_i = u + 1 + (q_i \bmod (N - u))$, где \bmod обозначает операцию взятия остатка от деления.

Формат выходного файла

Если это *обычный тест*, то выведите для каждого друга в отдельной строке единственное целое число — искомое максимальное время прибытия на конечную остановку в его маршруте.

Если это *тест-серия*, то выведите единственное целое число — остаток от деления суммы максимальных времен прибытия для всех друзей Иннокентия на 2^{32} .

Примеры

schedule.in	schedule.out
2 3	21
1	21
1 10 21	31
11 21 31	
1	
3	
1 2 1	
1 2 2	
1 2 3	
5 2	667
2 5 3 4	
1 3	
3 5	
10 11	
13 14	
18 23	
2	
9 10 5	

Комментарии

Приведем пояснение ко второму тесту из условия.

Это *тест-серия*. В нем у Иннокентия $5 \cdot 4 \cdot 2 = 40$ друзей. Например, с первой остановки вторым автобусом уезжают ровно пять друзей. Поясним, как в этом тесте для них определить конечные остановки.

$u = 1, w = 2$. Строим последовательность q_i : $q_1 = 9, q_2 = 10, q_3 = 1 \cdot 10 + 2 \cdot 9 + 42 = 70, q_4 = 1 \cdot 70 + 2 \cdot 10 + 42 = 132, q_5 = 1 \cdot 132 + 2 \cdot 70 + 42 = 314$.

По ней восстанавливаются конечные остановки для этих пяти друзей Иннокентия:

$v_1 = 1 + 1 + (9 \bmod 4) = 3, v_2 = 1 + 1 + (10 \bmod 4) = 4, v_3 = 1 + 1 + (70 \bmod 4) = 4, v_4 = 1 + 1 + (132 \bmod 4) = 2, v_5 = 1 + 1 + (314 \bmod 4) = 4$.

Система оценки

Тесты к этой задаче состоят из шести групп. Каждая группа, кроме нулевой, оценивается в 20 баллов. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп, исключая тесты из условия.

В группах тестов с первой по четвертую включительно вам предлагаются только *обычные тесты*.

0. Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
1. Тесты 3–12. В тестах этой группы $N = 2, M \leq 1000, Q \leq 1000$.
2. Тесты 13–22. В тестах этой группы $N = 2, M \leq 75\,000, Q \leq 75\,000$.
3. Тесты 23–37. В тестах этой группы $N \times M \leq 150\,000, N \times Q \leq 150\,000$.
4. В тестах этой группы $N \times M \leq 150\,000, Q \leq 150\,000$. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура.
5. В этой группе вам предлагаются только *тесты-серии*. Другие дополнительные ограничения отсутствуют. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура.

Задача Н. Силовое поле

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Империя обнаружила мятежников на ледяной планете Хот! По сведениям разведки все командование Альянса Повстанцев сейчас скрывается на базе «Эхо», спрятанной в горах на севере этой суровой планеты.

Для того, чтобы окончательно подавить силы восстания, необходимо в ходе стремительной атаки уничтожить эту базу и скрывающихся на ней мятежников. К сожалению, укрытие хорошо укреплено: в частности, его защищает мощное силовое поле, препятствующее бомбардировкам с орбиты. Силовое поле имеет форму выпуклого многоугольника с вершинами в N специальных станциях-ретрансляторах. Никакие три станции не располагаются на одной прямой.

Перед тем как начинать операцию по уничтожению повстанцев, требуется лишить их базу силового поля, уничтожив эти N станций точечным бомбометанием. Однако точные координаты этих станций нам неизвестны. Ваша цель — узнать расположение станций-ретрансляторов, чтобы наши войска смогли начать наступление.

На планете введена система координат, устроенная так, что все станции-ретрансляторы находятся в точках с целыми координатами, не превосходящими C по модулю.

В вашем распоряжении есть зонд-разведчик, оснащенный специальным оборудованием, позволяющим регистрировать станции-ретрансляторы. Если запустить его по прямой над базой повстанцев, по его информации можно будет узнать, сколько станций-ретрансляторов располагаются слева, и сколько — справа от прямой его движения. Станции, находящиеся на его пути, зонд не регистрирует.

С повстанцами надо расправиться как можно скорее: у вас есть время не более чем на 10^5 запусков этого зонда. Восстановите по полученной от него информации точные координаты станций-ретрансляторов, чтобы мы могли начать наступление, и Империя вас не забудет!

Формат взаимодействия с тестирующей системой

Это интерактивная задача.

При запуске решения на вход подаются два целых числа N и C — количество станций ($3 \leq N \leq 1000$) и ограничение на абсолютную величину их координат ($5 \leq C \leq 1000000$).

Для запуска зонда выведите строку «? $x_1 y_1 x_2 y_2$ », где (x_1, y_1) и (x_2, y_2) — две точки с целочисленными координатами, лежащие на прямой, по которой должен лететь зонд. Зонд будет лететь в направлении от первой точки ко второй. Точки не должны совпадать. Координаты точек не должны превосходить $5C$ по модулю.

На каждый запуск зонда-разведчика вводится полученная им информация — два целых числа l и r , разделенных пробелом, — количество станций-ретрансляторов слева и справа от прямой его движения соответственно.

Как только вы найдете ответ, выведите строку «Ready!» и в следующих N строках выведите координаты станций в любом порядке. После этого ваша программа должна завершиться.

Пример

ВВОД	ВЫВОД
4 5	? -1 3 1 3
0 4	? -1 2 1 2
0 3	? -1 1 0 2
0 3	? -1 0 0 2
0 2	? 0 0 0 2
1 1	? 1 0 1 2
3 1	? 2 0 2 2
3 0	? 3 0 1 2
3 0	Ready!
	0 -1
	2 1
	0 2
	-1 0

Комментарии

В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер вывода — для этого используйте `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.

Программа не должна делать более 10^5 запросов запуска зонда. При превышении этого количества, тест будет не пройден с вердиктом «Wrong Answer».

Во время тура вам будет доступна утилита `test.bat` или `test.sh`, с помощью которой вы сможете тестировать свои решения. Она имитирует работу проверяющей программы на заданных вами входных данных.

Чтобы протестировать свое решение, наберите в командной строке „`test.bat <solution> <test>`“ или „`bash test.sh <solution> <test>`“, где `<solution>` — исполняемый файл вашего решения, а `<test>` — файл, описывающий тест в следующем формате. В первой строке должны находиться два целых числа N и C , разделенных пробелом. В последующих N строках должны следовать координаты N станций-ретрансляторов *в порядке обхода против часовой стрелки*.

Утилита сообщит вам вердикт, который получило бы решение на этом тесте в тестирующей системе, и запишет в файл `<test>.log` полный протокол взаимодействия вашей программы и тестирующей.

Система оценки

Тесты к этой задаче состоят из четырех групп. Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

0. Тест 1. Тест из условия, оцениваемый в ноль баллов.
1. Тесты 2–11. В тестах этой группы $N = 3$, $C \leq 10$. Эта группа оценивается в 30 баллов.
2. Тесты 12–24. В тестах этой группы $N \leq 50$, $C \leq 100$. Эта группа оценивается в 30 баллов. Решение будет тестироваться на тестах этой группы только в случае прохождения всех тестов первой группы.
3. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline**, т.е. после окончания тура, причем только в случае прохождения всех тестов первой и второй групп.