

## Задача Eclair. Открытая олимпиада по дизайну

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

Если вы вдруг не знаете, то в подготовке соревнований по программированию обычно участвуют люди, которые в прошлом сами являлись участниками подобных соревнований. Это очень удобно: бывшие олимпиадники-программисты часто разбираются не только в написании программ, но и в вёрстке условий, и в настройке тестирующей системы, и ещё в множестве разных интересных (и не очень) дел. А что если бы олимпиаду готовили, например, дизайнеры?

В одной воображаемой вселенной проходит Открытая Олимпиада школьников по Дизайну, состоящая из  $n$  задач, которую готовят  $n$  дизайнеров условий, один дизайнер названий и один дизайнер шрифтов. Каждый из  $n$  дизайнеров условий уже сверстал условие своей задачи, но, так как дизайнеры люди творческие и работают поодиночке, каждый из них оставил разное количество места под название задачи. А именно, макет  $i$ -й задачи предполагает  $l_i$  букв в названии задачи.

Согласно регламенту олимпиады, требуется, чтобы названия задач состояли из символов Юникода, были **различны** и шли в лексикографическом порядке (см. замечание). Дизайнер шрифтов договорился с дизайнером названий, чтобы тот подобрал такие названия для задач, которые используют как можно меньше различных букв, и дизайнеру шрифтов пришлось бы работать над меньшим количеством изображений символов.

Пока дизайнер названий придумывает названия, дизайнер шрифтов решил поинтересоваться у вас, какое минимальное количество различных букв ему придётся нарисовать, чтобы из них можно было составить различные названия для всех задач и чтобы они шли в лексикографическом порядке. Считайте, что символов в Юникоде достаточно большое количество, чтобы это всегда можно было сделать для любых входных данных, удовлетворяющих ограничениям задачи. Обратите внимание, **не разрешается** менять порядок задач.

### Формат входных данных

В первой строке входных данных находится целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество задач в олимпиаде.

Во второй строке находится последовательность целых чисел  $l_1, l_2, \dots, l_n$  ( $1 \leq l_i \leq 10^9$ ) — длины названий задач.

### Формат выходных данных

Выведите единственное целое число — минимальное количество различных символов, которое потребуется использовать для составления названий для всех задач.

### Примеры

ВВОД	ВЫВОД
5 2 2 2 2 2	3
4 3 1 2 2	2

### Пояснение

Строка  $x_1x_2 \dots x_a$  длины  $a$  лексикографически меньше строки  $y_1y_2 \dots y_b$  длины  $b$ , если выполнено одно из двух условий:

- либо в первой позиции  $i$ , такой что  $x_i \neq y_i$ , в первой строке стоит меньший символ, чем во второй, то есть  $x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}, x_i < y_i$ ;
- либо первая строка является строгим префиксом второй строки, то есть  $a < b$  и  $x_1 = y_1, x_2 = y_2, \dots, x_a = y_a$ .

Последовательность различных слов называется упорядоченной лексикографически, если каждое слово в последовательности (кроме последнего) лексикографически меньше следующего слова.

В первом тесте из условия для составления названий задач можно воспользоваться символами «а» < «о» < «х» и составить названия «аа», «ао», «ах», «ох», «хх».

Во втором тесте из условия можно воспользоваться двумя символами «l» < «o» и составить названия «lol», «o», «ol» и «oo».

## Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп кроме тестов из условия. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. В данной задаче решение не обязано проходить тесты из условия, чтобы быть принятым на проверку.

Группа	Баллы	Дополнительные ограничения		Комментарий
		$n$	$l_i$	
0	0	–	–	Тесты из условия
1	11	$n \leq 10$	$l_i \leq 5$	Все $l_i$ одинаковые
2	7	$n \leq 10$	$l_i \leq 5$	
3	20	$n \leq 300$	$l_i \leq 300$	
4	20	$n \leq 5000$	$l_i \leq 5000$	
5	21	–	$l_i \leq 200\,000$	<b>Offline-проверка</b>
6	21	–	–	<b>Offline-проверка</b>

## Задача Muffin. Робот в поле

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

Любимое занятие Андрюши — компьютерные игры. Недавно он обнаружил новую очень интересную игру.

Мир игры представляет собой бесконечное поле, разбитое на клетки, а каждая клетка имеет свои внутриигровые координаты. Игрок управляет роботом, стоящим изначально в клетке с координатами  $(0, 0)$ . Нажимая на соответствующие клавиши клавиатуры, игрок может перемещать робота влево, вверх, вправо и вниз. Строго говоря, управление роботом происходит следующим образом:

- Нажав на кнопку «L», игрок уменьшит первую координату робота на 1.
- Нажав на кнопку «U», игрок увеличит вторую координату робота на 1.
- Нажав на кнопку «R», игрок увеличит первую координату робота на 1.
- Нажав на кнопку «D», игрок уменьшит вторую координату робота на 1.

Задача игрока состоит в том, чтобы привести робота в клетку с заданными координатами  $(x, y)$ . Кому-то из вас данная игра может показаться очень простой, но у Андрюши никак не получалось ее пройти. В конце концов, он совсем отчаялся и узнал у своего друга Жени, на какие клавиши и в каком порядке надо нажимать, чтобы пройти игру. Но все оказалось не так просто! Прохождение Жени не подошло Андрюше, видимо, тот перемещался в другую клетку, и он очень сильно разозлился! Он так сильно надавил на некоторые клавиши, что они сломались и перестали реагировать на нажатия. К своему удивлению Андрюша заметил, что эти поломки могут ему даже помочь!

Андрюша решил проверить, можно ли всё-таки пройти игру, воспользовавшись прохождением Жени и сломав некоторые клавиши между некоторыми нажатиями. Если какая-то клавиша сломана, то при последующих нажатиях этой клавиши робот не будет менять своего местоположения. Андрюша может сломать любую клавишу после некоторого нажатия, причем он может сломать несколько клавиш после одного и того же нажатия. Более того, Андрюша может сломать любые клавиши перед началом прохождения игры, если это поможет достичь заданной клетки. Даже если цель будет достигнута, Андрюша сломает оставшиеся клавиши, чтобы больше не возвращаться к этой игре.

Он попросил вас помочь ему справиться с этой задачей.

### Формат входных данных

В первой строке входных данных находится единственное целое  $n$  ( $1 \leq n \leq 1\,000\,000$ ) — количество ходов в Женином прохождении игры.

Следующая строка содержит  $n$  символов «L», «U», «R», «D» — описание прохождения.

В третьей строке содержится пара чисел  $x, y$  ( $-1\,000\,000 \leq x, y \leq 1\,000\,000$ ) — координаты клетки, в которой робот должен остановиться после всех нажатий.

### Формат выходных данных

Если никакие поломки клавиш не помогут Андрюше пройти игру, выведите единственное число  $-1$ .

В противном случае выведите четыре числа — номера нажатий, после которых нужно сломать клавиши «L», «U», «R», «D» соответственно. Если некоторую клавишу нужно сломать до начала игры, выведите для нее число 0. Так как Андрюша ломает оставшиеся целыми клавиши после игры, то вы должны вывести число в диапазоне от 0 до  $n$  для **каждой клавиши**.

Если существует несколько правильных вариантов ответа, выведите любой из них.

## Примеры

ВВОД	ВЫВОД
4 LRUD 3 2	-1
4 DURL 0 0	0 0 0 0
8 LLURDRRD 1 -1	1 7 6 8

## Пояснение

В первом тесте из условия клетка назначения находится слишком далеко от стартовой позиции, и добраться до неё за не более чем 4 нажатия невозможно.

Во втором тесте из условия можно сломать все кнопки с самого начала, тогда робот не сдвинется с места и автоматически окажется в требуемой позиции.

В третьем тесте из условия, если сломать кнопки, как указано в ответе, итоговая последовательность действий, проделанных роботом, будет выглядеть как LURDRD и приведёт его в требуемую позицию.

## Система оценки

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

Группа	Баллы	Дополнительные ограничения	Комментарий
		$n$	
0	0	–	Тесты из условия
1	32	$n \leq 20$	
2	29	$n \leq 500$	
3	39	–	

## Задача Napoleon. Застройка мегаполиса

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

В 2000 году Москва оказалась застроена до такой степени, что в черте города совсем не осталось территории, пригодной для постройки новых зданий. В поисках новых источников дохода правительство города приняло план, согласно которому все железнодорожные пути в черте города реконструируются и заменяются на подземные, а высвобожденная поверхность используется для коммерческой аренды.

Планирование будущей застройки было начато с участка Октябрьской железной дороги длиной в  $k$  метров. Так как строить здания поверх образованного туннеля — долгий и сложный процесс, было принято решение закрепить новый участок за наиболее популярными передвижными точками общепита, продающими мороженное, хот-доги, кофе и тому подобное.

Участок для строительства разделён на  $k$  сегментов одинаковой длины, последовательно пронумерованных целыми числами от 1 до  $k$ . Из  $n$  поступивших в правительство заявок на получение территории  $i$ -я претендует на сегменты с  $l_i$  по  $r_i$ , причём соответствующая точка общепита будет оказывать давление величиной  $p_i$  на соответствующий отрезок поверхности. Каждую заявку правительство либо отклонит, либо полностью удовлетворит, предоставив точке все запрошенные сегменты.

Правительство города заинтересовано в том, чтобы сдать каждый сегмент новообразованной территории в аренду хотя бы одной точке общепита. При этом, чтобы уменьшить риск обвала туннеля, было принято решение минимизировать максимальное из оказываемых давлений на каком-либо из сегментов. Обратите внимание, что не запрещается сдавать в аренду один сегмент сразу нескольким точкам общепита, но в таком случае давление, оказываемое ими на данный сегмент поверхности, суммируется.

Помогите правительству одобрить такой набор заявок, чтобы каждый сегмент был сдан в аренду хотя бы одной передвижной точке, но максимальное давление, оказываемое на туннель, было как можно меньше.

### Формат входных данных

В первой строке входных данных находятся два целых числа  $n$  и  $k$  ( $1 \leq n \leq 100\,000$ ,  $1 \leq k \leq 10^9$ ) — количество заявок на открытие точек общепита и количество сегментов поверхности.

В последующих  $n$  строках описаны заявки, каждая из которых задаётся тремя целыми числами  $l_i$ ,  $r_i$ ,  $p_i$  ( $1 \leq l_i \leq r_i \leq 10^9$ ,  $1 \leq p_i \leq 10^9$ ), соответственно границами предприятия и давлением, которое оно оказывает на поверхность туннеля.

### Формат выходных данных

Выведите наименьшее возможное максимальное давление, оказываемое на поверхность туннеля точками общественного питания, при условии, что все сегменты поверхности туннеля сданы в аренду. Если не существует способа покрыть весь участок, выведите  $-1$ .

## Примеры

ВВОД	ВЫВОД
3 4 1 3 1 3 4 2 1 4 5	3
1 3 1 2 1	-1
4 5 1 4 3 4 5 5 1 1 3 1 2 1	8
4 5 1 4 1 4 5 1 3 4 1 5 5 1	1

## Пояснение

В первом тесте из условия оптимальное решение — принять первые две заявки, тогда максимальное давление, равное 3, будет достигаться на третьем сегменте.

Во втором тесте из условия невозможно сдать в аренду третий сегмент.

В третьем тесте из условия одним из оптимальных решений будет удовлетворить все заявки, тогда максимальное давление, равное 8, будет достигаться на четвёртом сегменте. Обратите внимание, что минимизировать или максимизировать количество удовлетворённых заявок не требуется.

В четвёртом тесте из условия оптимальное решение — удовлетворить первую и четвёртую заявки, тогда на все сегменты будет оказываться одинаковое давление, равное 1.

## Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за группу ставятся только при прохождении всех тестов группы и всех групп, от которых зависит данная группа (см. таблицу с системой оценивания). **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. В данной задаче решение не обязано проходить тесты из условия, чтобы быть принятым на проверку.

Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		$n$	$p_i$		
0	0	—	—	—	Тесты из условия
1	8	$n \leq 10$	$p_i \leq 10^9$	—	
2	15	$n \leq 3000$	$p_i = 1$	—	
3	21	$n \leq 3000$	$p_i \leq 10^9$	1	
4	16	$n \leq 100\,000$	$p_i = 1$	2	
5	40	—	—	1 – 4	<b>Offline-проверка</b>

## Задача Panna Cotta. Поиграем?

Имя входного файла:	стандартный поток ввода
Имя выходного файла:	стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это **интерактивная** задача.

1804 год. Вице-президент Соединённых Штатов Аарон Бёрр вызывает на дуэль кандидата в губернаторы штата Нью-Йорк Александра Гамильтона за серию оскорбительных памфлетов в свой адрес.

Но Бёрр разумный человек и понимает, что даже если он убьёт Гамильтона на дуэли, он потеряет свою репутацию, и его политическая карьера будет закончена. Поэтому противники решили просто сыграть в игру. Для честности они решили сыграть в неё  $g$  раз.

Каждую игру Гамильтон загадывает целое положительное число  $n$ , а Бёрр пытается его отгадать. Для любого целого положительного  $x$  Бёрр может спросить у Гамильтона долю чисел между 1 и  $n$  включительно, которые делятся на  $x$ . Иными словами, спрашивая про  $x$ , он получает значение выражения

$$\frac{\lfloor \frac{n}{x} \rfloor}{n},$$

причём Гамильтон сообщает ему результат в виде **несократимой** дроби (здесь  $\lfloor r \rfloor$  обозначает результат округления вниз вещественного числа  $r$ ).

Помогите Бёрру найти ответ за некоторое заранее определённое число запросов.

### Формат взаимодействия с тестирующей системой

При запуске решения на вход подается целое число  $g$  — число игр между Гамильтоном и Бёрром ( $1 \leq g \leq 1000$ ).

Для каждого теста зафиксировано число  $q$  ( $6 \leq q \leq 60$ ) — максимальное количество запросов в одной игре. Гарантируется, что  $q$  запросов достаточно, чтобы решить задачу. Эти числа не сообщаются программе участника, но ограничения на эти числа в различных подзадачах приведены в таблице системы оценивания. Если программа участника делает более  $q$  запросов программе жюри, на этом тесте она получает в качестве результата тестирования «Wrong answer».

Чтобы сделать запрос, следует вывести строку «X  $t$ », где  $t$  — целое положительное число ( $1 \leq t \leq 10^{18}$ ), для которого требуется узнать значение выражения

$$\frac{\lfloor \frac{n}{t} \rfloor}{n}$$

В ответ на каждый запрос программа получает через пробел два числа  $a$  и  $b$  — числитель и знаменатель этой дроби **после сокращения** — или число  $-1$  в случае, если программа превысила ограничение по числу запросов.

Если программа определила загаданное число, то она должна вывести строку «N  $t$ », где  $t$  — предполагаемый ответ ( $1 \leq t \leq 10^{18}$ ). Если ответ был правильный, то в ответ программа получает строку «Correct», а если неправильный, то она получает строку «Wrong».

После этого программа переходит к следующей игре, если они остались, иначе она должна завершиться.

Обратите внимание, в случае считывания числа  $-1$  или строки «Wrong» вы **обязательно** должны сразу завершить вашу программу. В противном случае вердикт тестирующей системы может быть некорректным, в частности, вы можете получить вердикт Run-time error или Time limit exceeded!

Гарантируется, что в каждом тесте загаданные числа фиксированы в самом начале игры и не изменяются в зависимости от ваших запросов.

## Примеры

ВВОД	ВЫВОД
2	X 2
1 2	X 3
3 10	X 5
1 5	X 4
1 5	X 6
1 10	X 10
1 10	X 11
0 1	N 10
Correct	X 1
1 1	X 2
0 1	N 1
Correct	

## Замечания

В первом примере  $g = 2$ . Приведены примеры запросов, по которым игрок угадывает, что в первой игре загадано число 10, а во второй 1. Эти же числа загаданы в первом тесте в тестирующей системе.

В точности соблюдайте формат выходных данных. После каждого вывода обязательно выводите один перевод строки и 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.

## Система оценки

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

Группа	Баллы	Дополнительные ограничения	Комментарий
		$q$	
0	0	$q = 60$	Тесты из условия
1	30	$q = 60$	
2	10	$q = 30$	
3	4	$q = 20$	
4	4	$q = 15$	
5	5	$q = 10$	
6	5	$q = 9$	
7	11	$q = 8$	
8	10	$q = 7$	
9	21	$q = 6$	