

## Задача 1. Ваня и сложный шифр

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Ваня очень сильно любит читать книжки. Если у него есть хотя бы одна свободная минута, он тут же берется за книжку и начинает читать.

Однажды во время чтения одной из книг Ваня встретил довольно странное слово, состоящее из нулей и единиц. У этого слова была сноска, из которой Ваня понял, что это слово зашифровано. К счастью, в сноске было сказано, что это слово — закодированное число, каждая цифра в его записи заменена на соответствующий шифр. Шифры для каждой из цифр от 0 до 9 были даны. Известно, число однозначно декодируется, т.к. код, которым оно зашифровано, является префиксным. По определению, множество шифров формирует префиксный код, если любой шифр из этого множества не является началом любого другого шифра из этого же множества.

Ваня сумел справиться с расшифровкой числа, а для проверки просит вас сделать то же самое.

### Формат входных данных

В первой строке входного файла записана последовательность нулей и единиц — закодированное число из книжки.

В следующих десяти строках записаны по порядку коды цифр от 0 до 9, представляющие собой непустые последовательности из нулей и единиц.

Длина любого кода не превышает 5000 символов, длина закодированного числа — не более  $5 \cdot 10^5$  символов.

### Формат выходных данных

В выходной файл нужно вывести раскодированное число.

### Пример

<code>input.txt</code>	<code>output.txt</code>
111111110001011100010110111011100011	911442
110000	
1100010	
1100011	
11001	
1101	
11100	
11101	
11110	
111110	
111111	

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	40	Длины кодов не превышают 500 символов	1
3	60	Нет дополнительных ограничений	1, 2

## Задача 2. Иннокентий и химия

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда, для Java — 2 секунды
Ограничение по памяти:	256 мегабайт

Иннокентий — юный химик. Как у любого химика, у него есть банки с реактивами, которые разбросаны по его домашней импровизированной лаборатории.

Иннокентий заметил, что для очередного опыта ему приходится долго искать необходимые реактивы по всей лаборатории. Конечно же, ему это не нравится. Однажды по дороге из школы, он обратил внимание, что на витрине одного из магазинов выставлено специальное устройство для хранения реактивов. Устройство представляет собой стойку, на верхней панели которой просверлены на прямой небольшие отверстия, ведущие в отсеки достаточно большого объема, позволяющего налить в них любое количество реактива. Иннокентий незамедлительно приобрел это устройство.

Придя домой, Иннокентий решил перелить все свои реактивы в данное чудо химической техники. Чтобы сохранить свои реактивы невредимыми, Кеша переливал их в разные отсеки устройства. В процессе он задумался, насколько большим может быть расстояние между двумя самыми близкими отверстиями, ведущими к отсекам, содержащим реактивы? Помогите Иннокентию ответить на этот вопрос.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $N$  и  $M$ , где  $N$  — количество отсеков в устройстве,  $M$  — количество реактивов ( $2 \leq N \leq 2 \cdot 10^5$ ,  $2 \leq M \leq N$ ). В следующей строке записаны через пробел  $N$  различных целых положительных чисел — координаты отверстий на верхней панели устройства. Каждая из координат не превосходит  $10^9$ .

### Формат выходных данных

В первую строку выходного файла нужно вывести целое число — насколько большим может быть расстояние между двумя самыми близкими отверстиями, ведущими к отсекам, содержащим реактивы.

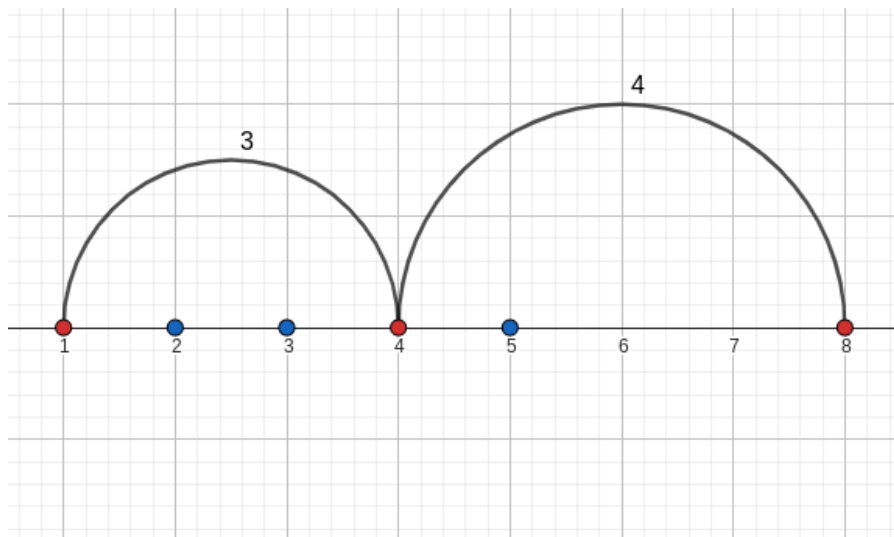
В следующую строку нужно вывести через пробел номера отсеков, в которые нужно перелить реактивы. Отсеки нумеруются с единицы в порядке перечисления их во входных данных.

Если существует несколько возможных вариантов разлить реактивы в сосуды согласно условиям задачи, то выведите любой из них.

### Пример

<code>input.txt</code>	<code>output.txt</code>
6 3 1 5 2 3 8 4	3 1 6 5

### Пояснение к примеру



Дуги заканчиваются в координатах отверстий, ведущих в те отсеки, в которые нужно перелить реактивы для достижения наибольшего минимального расстояния между ними.

## Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	20	$4 \leq N \leq 500, N - 2 \leq M \leq N$	1
3	20	$2 \leq N \leq 1000,$ координаты не превосходят $10^4$	1
4	60	Нет дополнительных ограничений	1, 2, 3

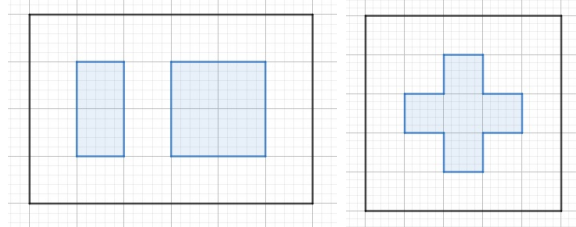
## Задача 3. Болотце

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 256 мегабайт

Каждая жабка хвалит свое болотце. Но сложно хвалить болотце, когда ты не знаешь в какой его точке находишься. Вот и наша знакомая Жабка решила перестраховаться перед тем, как начать хвалить.

Жабка проснулась в какой-то точке болота, но не знает в какой. План болота представляет собой прямоугольную клеточную область, каждой клетке которой присвоены координаты. Координаты  $(0, 0)$  имеет самая верхняя и левая клетка болота, а  $(N - 1, M - 1)$  самая нижняя и правая, соответственно. Жабка может перемещаться на плане в соседнюю клетку, у которой одна из координат отличается не более чем на единицу. Но не на все клетки Жабка может попасть — некоторые области болотца остаются недостижимыми для ее прыжков и подводных перемещений. Такие области Жабка называет препятствиями. Кроме того, Жабка не планирует покидать болотце и, когда достигает его края, также сообщает о том, что область за пределами болотца недостижима. Известно, что препятствия не касаются ни границ болота, ни друг друга, Жабка может обойти препятствие по воде с любой его стороны.

На болоте встречаются два вида препятствий — кочки и кувшинки. Кочки могут быть любого размера и представлены на плане болота, как прямоугольники. Кувшинки имеют определенную форму и занимают 5 клеток на плане болота:



Закрашенные области на левом рисунке — две кочки на плане болота, на правом рисунке изображена кувшинка на плане болота.

Несмотря на то, что Жабка плохо помнит, где проснулась, она точно помнит размеры болотца. Имея эту информацию и установив канал связи с Жабкой, помогите ей определить, в какой точке болотца она находится в данный момент. У Жабки есть некоторый запас сил, и нельзя допустить, чтобы она весь его потратила, иначе хвалить болотце будет некому.

## Протокол взаимодействия

Это интерактивная задача, и в ней вам предстоит работать не с файловым вводом-выводом, а со специальной программой — интерактором. Взаимодействие с ней осуществляется через стандартные потоки ввода-вывода.

При старте вашей программе в стандартный поток ввода подаётся два целых числа  $N$ ,  $M$ , где  $N$  — количество строк, а  $M$  — количество столбцов на плане болота ( $1 \leq N, M \leq 10^3$ ).

Ваша программа должна отправлять запросы в стандартный поток вывода.

Запрос должен содержать один из четырех символов:

- **L** — Жабка попытается переместиться влево (в случае успеха это уменьшит ее вторую координату на 1);
- **R** — Жабка попытается переместиться вправо (в случае успеха это увеличит ее вторую координату на 1);
- **U** — Жабка попытается переместиться вверх (в случае успеха это уменьшит ее первую координату на 1);
- **D** — Жабка попытается переместиться вниз (в случае успеха это увеличит ее первую координату на 1).

В ответ на запрос приходит один из вариантов ответа:

- **Success** — Жабка успешно переместилась на соседнюю клетку;
- **Impossible** — на клетку невозможно переместиться (она находится за границами болотца или является недостижимым препятствием).

Как только вы поняли, в какой точке **в данный момент** находится Жабка, нужно вывести запрос с ее координатами в формате  $X \ a \ b$ , где  $a$  — номер клетки по горизонтали,  $b$  — номер клетки по вертикали. После этого ваша программа должна завершить работу. Если введенные координаты не совпадают с действительными, ваше решение получает вердикт **Wrong Answer**.

Если после  $2 \cdot (N + M)$ -го запроса получен ответ, отличный от координат Жабки, решение получает вердикт **Wrong Answer**.

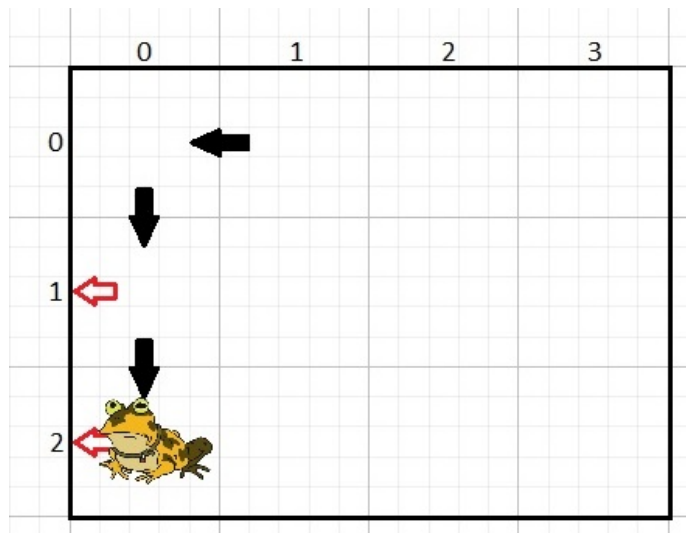
Убедитесь, что вы выводите символ перевода строки и очищаете буфер потока вывода (команда **flush** языка) после каждого выведенного запроса. Иначе решение может получить вердикт **Timeout**.

## Пример

стандартный ввод	стандартный вывод
3 4	-
-	L
Success	-
-	D
Success	-
-	L
Impossible	-
-	D
Success	-
-	L
Impossible	-
	X 2 0

## Пояснение к примеру

На картинке черными закрашенными стрелками отмечены успешные перемещения Жабки, а красными контурами стрелок — неуспешные.



## Система оценки

Количество баллов, которое получит ваше решение, зависит от сложности передвижения по болоту. Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	15	На болоте нет препятствий	1
3	15	Встречаются только кочки	1, 2
4	30	Встречаются только кувшинки	1, 2
5	40	Могут встретиться оба вида препятствий	1, 2, 3, 4

## Задача 4. Треугольники

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Вова любит искать структуру в любых случайных объектах. Однажды, глубоко задумавшись, он стал рисовать точки на прямоугольном листе. Посмотрев на полученный результат, он задумался ещё больше. Вова быстро прикинул, сколько существует равнобедренных треугольников с вершинами в заданных точках. Однако его интересуют не все треугольники, а только содержащие, как минимум, одну сторону определенной длины и имеющие определенную площадь. Вова считает, что именно в них ключ к загадкам мироздания. Но у него возникли сложности с вычислением количества таких треугольников, и он попросил вас помочь.

### Формат входных данных

В первой строке входного файла записаны целые числа  $N$ ,  $t$  и  $S$  — количество точек, требуемая длина стороны и площадь соответственно ( $1 \leq N \leq 10^5$ ,  $1 \leq t \leq 10^5$ ,  $1 \leq S \leq 10^9$ ).

Каждая из  $N$  последующих строк содержит пару целых чисел  $x_i, y_i$  — координаты точек на листе, считая от левого верхнего угла ( $0 \leq x_i, y_i \leq 10^5$ ). Гарантируется что все точки имеют различные координаты.

### Формат выходных данных

В выходной файл требуется вывести одно целое число — количество равнобедренных треугольников с вершинами в заданных точках, таких что хотя бы одна сторона имеет длину  $t$ , а площадь равна  $S$ .

### Пример

input.txt	output.txt
4 2 2 0 0 2 0 2 2 0 2	4

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	23	$1 \leq N \leq 100$	1
3	36	$1 \leq N \leq 5000$	1, 2
4	41	$1 \leq N \leq 10^5$	1, 2, 3

# Задача 5. Задачи на геометрию

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда, для Java — 2 секунды  
Ограничение по памяти: 256 мегабайт

Артём, большой любитель сложных задач, последовательностей и геометрии, составляет задачи для  $N$  соревнований по спортивному программированию.

Артём считает, что одним из важнейших качеств соревнований является сбалансированность. Для простоты модели каждое соревнование описывается одним числом — количеством задач на геометрию, при этом неважно, что именно это за задачи.

Для того, чтобы последовательность из  $N$  соревнований была сбалансированной, должны выполняться два условия:

1. В каждом соревновании должно быть не больше  $M$  задач на геометрию.
2. Количества задач на геометрию в двух последовательных соревнованиях должны отличаться не более, чем на  $K$  задач.

Артёму интересно, сколькими способами он может создать сбалансированную последовательность соревнований.

## Формат входных данных

В первой строке входного файла содержится три целых числа  $N$ ,  $M$  и  $K$  — число соревнований по программированию, максимальное число задач на геометрию, разрешенное в одном соревновании и максимальное число задач на геометрию, на которое могут отличаться последовательные соревнования ( $0 < N \leq 10^{18}$ ,  $0 \leq M \leq 200$ ,  $0 \leq K \leq M$ ).

## Формат выходных данных

В выходной файл необходимо вывести одно число — количество возможных сбалансированных последовательностей соревнований.

Так как это число может быть очень большим, требуется вывести его по модулю  $10^9 + 7$ .

## Примеры

input.txt	output.txt
200 100 0	101
10 3 3	1048576
5 3 1	178
1234 56 7	302091741

## Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тест из условия	
2	11	$0 < N \leq 10^{18}$ , $0 \leq M = K \leq 200$	1
3	23	$N \cdot M \cdot K \leq 5 \cdot 10^7$	1
4	29	$N \cdot M \leq 5 \cdot 10^7$	1, 3
5	37	$0 < N \leq 10^{18}$ , $0 \leq M \leq 200$ , $0 \leq K \leq M$	1, 2, 3, 4