

## Задача 1. Толя и робот

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Толе на Новый год подарили крутого робота! Его главная особенность заключается в том, что робот понимает простые голосовые команды, такие как «Вверх», «Вниз», «Налево» и «Направо». А еще, робота можно программировать и учить сложным маневрам!

Как и бывает с любой новой игрушкой: ее хочется сразу испытать. Толя придумал несложную задачку для робота: он поместит его на прямоугольное клеточное поле так, чтобы робот находился внутри этого поля, но не на его границе. После этого будут даваться команды роботу. Робот должен уметь передвигаться по клеткам и отслеживать свое местонахождение в поле. Он должен сообщать – находится ли он строго внутри поля или на его границе. А в случае, если очередная команда Толи выведет игрушку за границу поля, то робот должен сообщить об этом и выключиться.

Осталась одна проблема – Толя плох в программировании на языках высокого уровня, поэтому просит вас помочь ему с написанием алгоритма для робота.

### 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 четыре целых числа: размеры поля – количество строк и столбцов, соответственно, и координаты начального положения робота ( $3 \leq N \leq 10^5$ ,  $3 \leq M \leq 10^5$ ,  $1 < X < N$ ,  $1 < Y < M$ ). Левая верхняя клетка на плане поля имеет координаты (1, 1).

Ваша программа должна сообщать о том, где находится робот с помощью запросов в стандартный поток вывода.

Запросы могут быть следующими: `inside ?` или `border ?`.

Запрос первого типа мы делаем, если находимся строго внутри поля, второго – на одной из граничных клеток.

Ответом на эти запросы является одна из четырех команд: `U`, `D`, `L` или `R`, что означает соответственно «Вверх», «Вниз», «Налево» и «Направо». Как только робот вынужден будет покинуть поле, он должен вывести сообщение `End of program` и закончить работу.

Гарантируется, что запросов на изменение положение робота не более, чем  $2 \cdot 10^5$ .

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`.

## Примеры

стандартный ввод	стандартный вывод
4 5 2 4 R D L D D	inside ? border ? border ? inside ? border ?
3 3 2 2 U L D D R R U U L D D D	inside ? border ? border ? border ? border ? border ? border ? border ? border ? border ? border ? inside ? border ?

## Задача 2. Вадим и столовая

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вадим больше всего на свете любит плотно поесть. Но, как и любой другой школьник, он ограничен в средствах, и поэтому отказывает себе в удовольствии есть все подряд. Однако, если он очень сильно захочет съесть какое-нибудь блюдо, то не успокоится, пока не получит его.

Сегодня по пути в столовую Вадим составил у себя в голове список блюд, которые он непременно хотел бы купить, и уже прикинул, сколько денег у него уйдет на обед. Каково же было его удивление, когда он узнал, что в его любимой столовой появилась новая акция: комплексные обеды!

Работает эта акция невероятно просто. Каждый комплексный обед задается списком блюд, которые в него входят, а также ценой, которая может оказаться меньше той, которую заплатил бы посетитель столовой, покупая блюда из комплекса по отдельности (но это не гарантируется).

Вадим чувствует, что может сэкономить, но при этом он не хочет отклоняться от намеченного плана. Он намерен взять все блюда, которые есть в его мысленном списке желаний, но заплатить при этом как можно меньше денег. Вадим готов даже взять лишние блюда, если это окажется выгодно. Обратите внимание, что Вадим может купить несколько комплексных обедов. Помогите ему составить заказ.

### Формат входных данных

В первой строке входного файла задано число  $n$  — количество блюд в мысленном списке желаний Вадима ( $1 \leq n \leq 200$ ).

В следующих  $n$  строках задается список желаний. В каждой строке через пробел записаны два числа:  $a_i$  и  $c_i$  — уникальный номер  $i$ -го блюда из списка и его цена, если покупать его отдельно ( $1 \leq a_i, c_i \leq 10^9$ ). Блюда в списке не повторяются.

В следующей ( $n + 2$ )-й строке задано число  $m$  — количество комплексных обедов в столовой ( $1 \leq m \leq 20$ ).

В следующих  $2m$  строках описываются комплексные обеды. Каждый обед задается двумя строками, идущими подряд.

В первой строке задаются два числа  $k_j$  и  $c_j$  — количество блюд в  $j$ -м обеде и цена  $j$ -го обеда соответственно ( $1 \leq k_j \leq 200$ ,  $1 \leq c_j \leq 10^9$ ).

Во второй строке записаны  $k_j$  чисел:  $a_{j1}, \dots, a_{jk_j}$  — номера блюд, входящих в  $j$ -й обед ( $1 \leq a_{jk_l} \leq 10^9$ ). Некоторые из блюд могли не встречаться в списке Вадима. Каждое блюдо может встретиться в обеде максимум один раз.

### Формат выходных данных

В выходной файл необходимо вывести одно целое число — минимальную сумму, за которую Вадим сможет купить все блюда из своего списка желаний.

### Система оценки

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

1.  $n \leq 10$ ,  $m \leq 10$ , все  $k_j \leq 10$  (**20 баллов**);
2.  $n \leq 20$ ,  $m \leq 20$ , все  $k_j \leq 20$  (**60 баллов**);
3.  $n \leq 200$ ,  $m \leq 20$ , все  $k_j \leq 200$  (**100 баллов**).

**Пример**

input.txt	output.txt
6 100 5 200 5 300 5 400 5 500 5 600 3 2 4 6 100 200 300 700 3 5 300 400 500	14

**Замечание**

В примере из условия Вадиму выгоднее всего взять оба комплексных обеда и купить блюдо с номером 600 отдельно.

## Задача 3. Лиза и НОДы

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

Девочка Лиза учится в начальной школе, но, несмотря на свой возраст, уже знает, что такое наибольший общий делитель (НОД). Более того, если вы дадите Лизе два натуральных числа, она без всяких затруднений сможет найти их НОД.

До недавнего времени она этим и занималась: выписывала на листочек произвольные натуральные числа и находила их НОДы. Однако, ей быстро наскучило это занятие, и тогда она придумала новую игру.

Ее правила таковы: Лиза записывает в двух строках  $2n$  натуральных чисел, по  $n$  чисел в каждой строке. Затем она каким-то образом перемешивает числа в каждой строке, не перенося числа из одной строки в другую. После этого заполняет третью строку следующим образом: на  $i$ -ю позицию третьей строки она записывает НОД двух чисел, которые после перемешивания оказались на  $i$ -й позиции в первой и второй строках.

Очевидно, что набор чисел в третьей строке будет зависеть от того, как Лиза перемешивает числа в первых двух строках. Наша юная героиня задалась вопросом: насколько большой может оказаться сумма чисел в последней строке? К сожалению, она еще не научилась программировать, и самостоятельно найти ответ на этот вопрос не может. Поэтому она просит вашей помощи.

### Формат входных данных

В первой строке входного файла записано натуральное число  $n$  ( $1 \leq n \leq 200$ ).

Во второй и третьей строках записаны  $2n$  натуральных чисел, по  $n$  чисел в каждой строке:  $a_1, \dots, a_n, b_1, \dots, b_n$  — это числа, которые Лиза записала на свой листочек в начале игры ( $1 \leq a_i, b_i \leq 10^{16}$ ).

### Формат выходных данных

В выходной файл необходимо вывести одно целое число — максимальную сумму, которую можно получить из чисел в третьей строке в конце игры.

### Система оценки

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

1.  $n \leq 8$  (20 баллов);
2.  $n \leq 15$  (40 баллов);
3.  $n \leq 200$  (100 баллов).

### Примеры

input.txt	output.txt
3 2 3 4 6 8 9	9
5 1 3 5 7 9 15 20 30 40 50	13

### Замечание

Пояснение ко второму примеру.

Максимальную сумму НОДов мы можем получить, например, следующим перемешиванием:

3	9	5	7	1
15	30	50	20	40
3	3	5	1	1

## Задача 4. Кеша и выражения

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

Недавно на уроке математики Иннокентий узнал про задание приоритета арифметических операций в выражениях. Быстро сделав домашнее задание, он задумался, сколько всего различных выражений длины  $n$  существует. Помогите ему!

Определим выражения следующим образом:

- $x$  — выражение.
- Если  $S$  — выражение, то  $(S)$  также выражение;
- Если  $S_1$  и  $S_2$  выражения, то  $S_1 + S_2$  и  $S_1 - S_2$  — выражения.

### Формат входных данных

Во входном файле задано число  $N$  — длина выражения ( $0 \leq N \leq 10^6$ ).

### Формат выходных данных

Выходной файл должен содержать одно целое число — количество различных выражений длины  $N$ . Так как таких выражений может быть много, то нужно вывести остаток от деления точного ответа на число 998244353.

### Система оценки

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

- $N \leq 20$  (**20 баллов**);
- $N \leq 4000$  (**60 баллов**);
- $N \leq 10^6$  (**100 баллов**).

### Примеры

input.txt	output.txt
3	3
5	11

### Замечание

Возможные выражения длины 3:

$(x)$   
 $x + x$   
 $x - x$

Возможные выражения длины 5:

$((x))$   
 $(x) + x$   
 $(x) - x$   
 $(x + x)$   
 $(x - x)$   
 $x + (x)$   
 $x + x + x$   
 $x + x - x$   
 $x - (x)$   
 $x - x + x$   
 $x - x - x$

## Задача 5. Угадай прямую

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Учитель по геометрии Василий Петрович придумал следующую игру для своих школьников, чтобы они лучше усвоили азы стереометрии. Сначала он загадывает несколько прямых в пространстве. Затем школьники начинают предлагать различные плоскости, а в ответ получают от учителя точки пересечения загаданных прямых с этими плоскостями. Цель школьников – разгадать все прямые, т.е. для каждой прямой указать две различные точки, которые на ней лежат.

На первой неделе школьникам угадать прямые не удалось, поэтому Василий Петрович написал программу для удалённого угадывания прямых. Теперь они могут практиковаться и тренироваться в любое время. Школьники бьются уже несколько недель и просят Вас написать программу, которая поможет им угадать все прямые Василия Петровича.

### 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.

В начале вашей программе в стандартный поток ввода подаётся одно целое число  $N$  – количество загаданных прямых ( $1 \leq N \leq 10^4$ ).

Ваша программа может делать запросы о пересечениях загаданных прямых с плоскостями. Предполагается, что каждая плоскость задаётся уравнением вида  $ax + by + cz + d = 0$ . Каждый запрос состоит из четырех целых чисел  $a, b, c, d$ , одновременно неравных нулю – коэффициентов уравнения плоскости, разделённых символом пробела ( $|a|, |b|, |c|, |d| \leq 10^3$ ).

Ответом на каждый запрос является  $N$  строк, соответствующих пересечениям плоскости, заданной уравнением  $ax + by + cz + d = 0$ , с исходными прямыми. Каждая строка либо содержит три вещественных числа  $x, y, z$  – координаты единственной точки пересечения соответствующей прямой с плоскостью, либо строку `parallel`, если прямая параллельна данной плоскости. Координаты выводятся с семью знаками после десятичной точки.

Ваша задача угадать все прямые за не более чем 10 запросов. Если прямые отгаданы, то вместо запроса Вы должны вывести строку `answer:`, а затем  $N$  строк, состоящих из шести вещественных чисел  $x_1, y_1, z_1, x_2, y_2, z_2$  таких, что точки  $(x_1, y_1, z_1)$  и  $(x_2, y_2, z_2)$  лежат на соответствующей прямой. Все координаты не должны превышать  $10^9$  по модулю. Гарантируется, что каждая из загаданных прямых проходит через две точки, координаты которых целые и не превышают  $10^3$  по модулю.

Относительная или абсолютная погрешность ответа не должна превышать  $10^{-6}$ .

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
2	1 1 1 0
0.000000 0.000000 0.000000	1 1 1 1
parallel	1 2 3 4
-0.3333333 -0.3333333 -0.3333333	4 3 2 1
parallel	1 1 0 1
-0.6666667 -0.6666667 -0.6666667	1 0 1 0
parallel	answer:
-0.1111111 -0.1111111 -0.1111111	0.0 0.0 0.0 1.0 1.0 1.0
parallel	4.0 -5.0 1.0 1.0 1.0 -2.0
-0.5000000 -0.5000000 -0.5000000	
4.0000000 -5.0000000 1.0000000	
0.0000000 0.0000000 0.0000000	
1.5000000 0.0000000 -1.5000000	



## Задача 6. Игра со строкой

Имя входного файла: `input.txt`  
 Имя выходного файла: `output.txt`  
 Ограничение по времени: 3 секунды  
 Ограничение по памяти: 256 мегабайт

Мише и Саше стало скучно, и они придумали следующую игру: Миша пишет на листке строку  $S$ , после чего просит Сашу произвести над ней  $Q$  операций двух видов:

1. *set*  $l r c$  — заменить все символы подстроки, начинающейся с позиции  $l$  и заканчивающейся в позиции  $r$ , на символ  $c$ .
2. *ask*  $l r$  — узнать, является ли подстрока, начинающаяся с позиции  $l$  и заканчивающаяся в позиции  $r$ , палиндромом.

Так как Саша устал повторять эти действия, он просит вас написать программу, которая всё будет делать за него.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $N$  и  $Q$  — длину загаданной Мишей строки и количество операций ( $1 \leq N \leq 10^5$ ,  $1 \leq Q \leq 2 \cdot 10^5$ ).

Во второй строке написана сама строка  $S$ . Она содержит только маленькие латинские буквы. Нумерация символов в строке  $S$  начинается с единицы.

Далее идут  $Q$  строк, каждая из которых может быть операцией одного из двух видов:

`set l r c` или `ask l r` ( $1 \leq l \leq r \leq N$ ).

### Формат выходных данных

Для каждой операции вида `ask` в выходной файл необходимо вывести в новую строку слово `YES`, если подстрока является палиндромом, и строку `NO`, в противном случае.

### Система оценки

В задаче две группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

1.  $n \leq 2 \cdot 10^3$ ,  $q \leq 5 \cdot 10^3$  (20 баллов);
2.  $n \leq 10^5$ ,  $q \leq 2 \cdot 10^5$  (100 баллов);

### Пример

input.txt	output.txt
5 5	NO
abcde	YES
ask 1 5	YES
set 2 4 z	
ask 2 4	
set 5 5 a	
ask 5 5	

### Замечание

Палиндром — строка, которая равна себе, записанной в обратном порядке.

## Задача 7. Шахматинатор

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Простая официантка Кара Соннор точно знает: из будущего скоро придёт машина, посланная за ней восставшим искусственным интеллектом ChessNet. Сам ChessNet коварен и опасен, но так как этот ИИ изначально создавался для игры в шахматы, он пока сумел построить только машину «Шахматинатор-3000».

Эта машина умеет имитировать ходы любой шахматной фигуры. По умолчанию действуя в режиме пешки, Шахматинатор может потратить несколько единиц времени, чтобы переключиться на режим движения какой-либо другой шахматной фигуры. Разные фигуры требуют разное количество времени для активации, но один ход любой фигуры занимает у Шахматинатора одну единицу времени.

Повстанцы из будущего передали Каре всю информацию по Шахматинатору, которая у них есть, в том числе и точные координаты точки в городе, куда будет произведена его высадка из машины времени. Теперь Каре нужно найти место, где лучше всего спрятаться, чтобы дождаться подкрепления повстанцев. К сожалению, разведывательные технологии будущего шагнули далеко вперед, и, где бы Кара ни пряталась, Шахматинатор сразу узнает её точные координаты, едва придёт в её время. Поэтому ей нужно найти такую точку на карте города, куда «Шахматинатор-3000», действуя оптимально, будет добираться дольше всего, и дождаться там подкрепления.

Карта города представляется в виде дискретного прямоугольного поля с квадратными клетками. Размер клетки таков, что и Шахматинатор, и Кара могут разместиться в одной из них. Каждая клетка нумеруется по строкам сверху вниз, а по столбцам слева направо, соответственно, самая левая и верхняя клетка имеет координаты  $(1, 1)$ .

Некоторые клетки на карте заняты стенами, и Шахматинатор не может проходить сквозь них. Поэтому он, имитируя ход шахматной фигуры, заканчивает его, когда встречает препятствие. Исключение составляет лишь ход конем, так как конь может перепрыгивать препятствия. Никакой ход Шахматинатора не может закончиться в клетке со стеной, и он не может выходить за пределы карты. Ниже в Замечаниях приведены правила хода всех шахматных фигур, которые использует «Шахматинатор-3000».

Так как Кара Соннор просто официантка, а не программист, она обратилась к вам за помощью. Напишите программу, которая позволит ей найти нужное место, чтобы спрятаться и успеть спасти человечество.

### Формат входных данных

В первой строке входного файла записаны четыре целых числа: размеры поля в виде количества строк  $N$  и столбцов  $M$  и координаты клетки, в которую придёт Шахматинатор:  $X$ ,  $Y$  ( $1 \leq N, M \leq 300$ ,  $1 \leq X \leq N$ ,  $1 \leq Y \leq M$ ).

Во второй строке входного файла содержится шесть целых чисел – количество единиц времени, которое нужно потратить Шахматинатору, чтобы переключиться в режим пешки, коня, ладьи, слона, ферзя и короля соответственно. Количество единиц времени на переключение режимов не превышает  $10^6$ . При появлении в стартовой клетке Шахматинатор уже находится в режиме пешки.

В следующих  $N$  строках содержится карта местности, где Кара собирается спрятаться, по  $M$  символов в каждой строке. Если клетка пустая, то она обозначается символом '.', а если содержит стену, то символом '#'.

### Формат выходных данных

В единственную строку выходного файла необходимо вывести три целых числа – координаты клетки, куда необходимо спрятаться Каре для того, чтобы путь Шахматинатора был самым долгим, и количество единиц времени, которое займет этот путь у него.

Может случиться так, что на поле существует клетка, которая является недостижимой из начального положения Шахматинатора. В таком случае необходимо сообщить координаты такой клетки и

вместо третьего числа через пробел написать слово `INF`.

В случае, если клеток, подходящих в качестве ответа, больше одной, то необходимо вывести клетку с минимальным номером строки и столбца.

### Система оценки

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

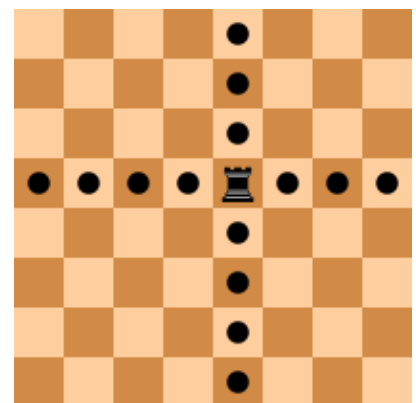
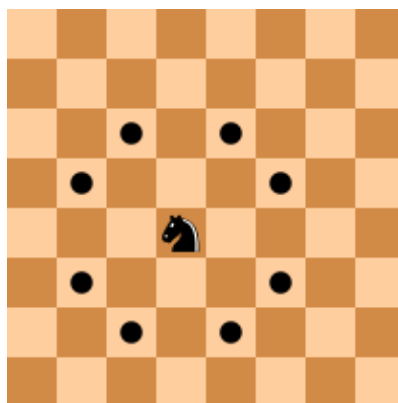
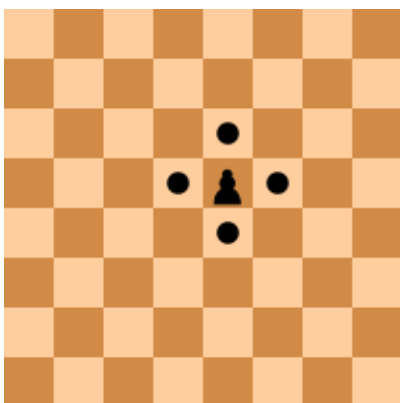
1.  $N, M \leq 18$  (20 баллов);
2.  $N, M \leq 70$  (60 баллов);
3.  $N, M \leq 300$  (100 баллов).

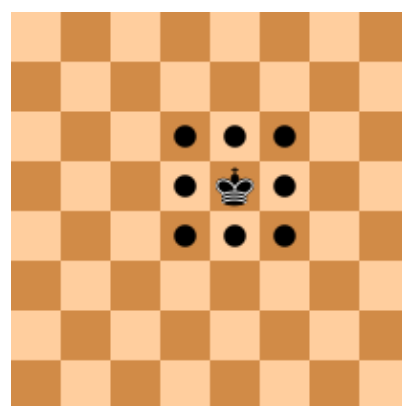
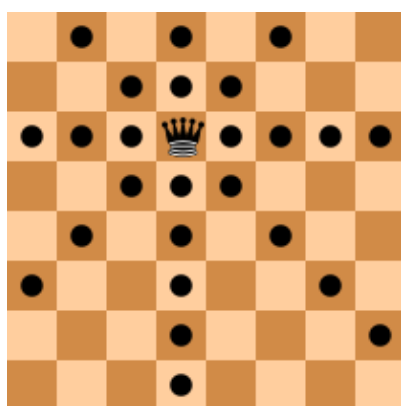
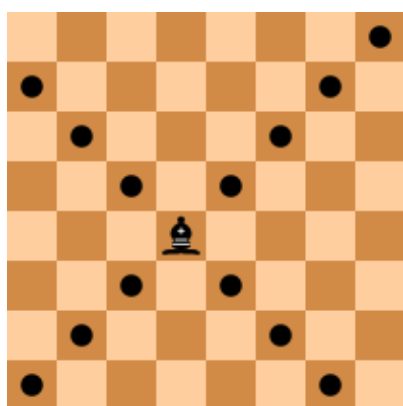
### Примеры

input.txt	output.txt
<pre>2 2 1 1 1 1 1 1 1 1 .. ..</pre>	<pre>2 2 2</pre>
<pre>3 4 2 2 1 2 3 4 5 6 ..#. ..#. ....</pre>	<pre>2 4 4</pre>
<pre>3 5 2 2 1 1 1 1 1 1 .###. .###. .###.</pre>	<pre>1 5 INF</pre>

### Замечание

Слева направо, сверху вниз: возможные ходы пешки, коня, ладьи, слона, ферзя и короля, соответственно.





## Задача 8. Виталья и карта сокровищ

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы когда-нибудь находили настоящую карту сокровищ? А Виталья нашёл! Оказалось, что несметные богатства закопаны на некотором острове, название которого мы, разумеется, раскрывать не будем. Впрочем, несколько деталей вам всё-таки следует знать. Этот остров имеет форму прямоугольника, и карту острова можно представить себе, как тетрадный лист в клетку. Каждая клетка имеет координату  $(x, y)$ , где  $x$  — номер строки,  $y$  — номер столбца. Левая верхняя клетка имеет координату  $(1, 1)$ .

Известно, что сокровища закопаны в месте, которому на карте соответствует одна из клеток. На обратной стороне карты записаны инструкции, как эту клетку найти. Сначала оговаривается, какую клетку мы считаем стартовой, даются координаты этой клетки. Далее следует строка, состоящая из команд, указывающих, в каком направлении нам нужно сдвинуться на одну клетку из текущей: вправо, влево, вверх или вниз. Выполняя поочерёдно все эти команды, начиная со стартовой клетки, в итоге мы окажемся в клетке с сокровищами.

Казалось бы, ничего сложного. Однако некоторые трудности всё же возникли. Карта оказалась очень старой, и поэтому некоторые символы в строке команд стёрлись, и распознать их не удалось. Теперь Виталья не может сказать наверняка, в какой клетке зарыт клад, однако он преисполнен желания перебрать все возможные варианты! Он просит вас помочь определить для каждой клетки острова, может ли клад быть зарыт в этой клетке. Считаем, что клад может быть зарыт в данной клетке, если на место пропущенных команд можно корректно подставить некоторые команды так, чтобы карта сокровищ вела в данную клетку. Корректная подстановка команд означает, что при выполнении данных команд мы ни разу не выйдем за границы острова.

### Формат входных данных

В первой строке входного файла через пробел заданы числа  $n, m, x, y$  — размеры острова и координаты стартовой клетки ( $1 \leq n, m \leq 200, n \cdot m \geq 2, 1 \leq x \leq n, 1 \leq y \leq m$ ).

Вторая строка состоит из символов «R», «L», «U», «D», «?» и представляет собой набор команд. Символ «?» означает, что на этом месте мог стоять любой из символов «R», «L», «U», «D». Гарантируется, что длина строки не превышает  $2 \cdot 10^5$ , строка не может быть пустой, и количество символов «?» в строке не превышает 1000.

### Формат выходных данных

В выходной файл необходимо вывести таблицу размера  $n \times m$ , состоящую из символов «1» и «0» (без пробелов внутри каждой строки), где «1» на позиции  $(i, j)$  означает, что в клетке  $(i, j)$  может быть зарыт клад, а «0» означает, что не может.

Гарантируется, что до того, как некоторые символы в карте стёрлись, набор команд был корректным. Это означает, что существует хотя бы одна клетка, в которой клад может быть зарыт.

### Система оценки

Пусть  $s$  — длина строки с командами,  $q$  — количество символов «?» в этой строке.

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

1.  $s \leq 1000, q \leq 10$  (20 баллов);
2.  $s \leq 1000, q \leq 1000$  (40 баллов);
3.  $s \leq 2 \cdot 10^5, q \leq 1000$  (100 баллов).

## Примеры

input.txt	output.txt
3 3 1 1 R?DDU	000 101 000
3 4 2 2 UR?R?	0101 0010 0001

## Замечание

Рассмотрим первый пример. В строке команд у нас есть один символ «?». Если мы на его место подставим «*U*», то мы сразу же выйдем за границу. Если подставим «*D*», то выйдем за границу после 4-го хода. Если же подставим «*L*» или «*R*», то в конце окажемся в клетках (2,1) или (2,3) соответственно. Именно на этих позициях в таблице стоят символы «1».