

## Problem A. Appearance Analysis

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 512 mebibytes

У Вас есть фотография здания факультета электротехники и вычислительной математики в Загребе. Овна в этом здании имеют одинаковый размер и расположены в виде прямоугольника  $r$  строк на  $c$  столбцов. Каждое окно, в свою очередь, является прямоугольником; все окна имеют одинаковую размерность. Каждый единичный квадрат в окне является или прозрачным (обозначается символом “.”) или окрашенным (обозначается символом “+”). Два окна имеют одинаковый дизайн, если одно из них можно повернуть на угол, кратный 90 градусам, так, чтобы прозрачные квадраты соответствовали прозрачным квадратам другого окна, а окрашенные — окрашенным. Отражать окно относительно вертикальной оси при этом *нельзя*.

На фасаде здания ряды и столбцы окон разделены одним слоем кирпича (обозначаемым символом “#”). Более формально, между каждыми двумя рядами окон, а также перед первым окном и после последнего окна расположен один ряд символов “#”. Аналогично, между каждыми двумя столбцами окон, а также перед самым левым и после самого правого расположен один столбец символов “#”.

Найдите количество различных дизайнов окон на заданной фотографии.

### Input

Первая строка входа содержит два целых числа  $r$  и  $c$  ( $3 \leq r, c \leq 111$ ) — количество строк и столбцов на фотографии. Далее задана сама фотография —  $r$  строк, каждая из которых содержит по  $c$  символов в соответствии с условием задачи.

Гарантируется, что фотография задана корректно и что окна имеют ненулевую площадь.

### Output

Выведите одно целое число — количество различных дизайнов окон на фотографии.

## Examples

standard input	standard output
<pre> 11 16 ##### #...#++++#...# #...#++.#+...# #...#.++.#++.+# #...#...#++++# ##### #...#.+.#+++++# #..++#.+.#++.+# #+...#...#.++.# #+...#.++#.++.# #####                     </pre>	<pre> 4                     </pre>
<pre> 9 21 ##### #...+#+++++#...#...# #..+.#+.++.#+...#...# #+...#...#...#...# ##### #+...#...#...#...# #..+.#+.++.#+...#...# #...+#+++++#...#...# #####                     </pre>	<pre> 4                     </pre>

## Problem C. Convex Contour

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 512 mebibytes

Некоторое количество геометрических объектов расположены на квадратной решётке так, что объекты занимают последовательные клетки в одном ряду, и одна клетка содержит ровно один объект. Каждый объект может быть:

- или квадратом, совпадающим с квадратом решётки,
- или кругом, вписанным в квадрат решётки,
- или правильным треугольником, у которого одна из сторон совпадает с нижней стороной квадрата решётки.

На иллюстрациях снизу можно увидеть объекты из первого примера и их *выпуклый контур*.



Неформально, *выпуклый контур* — это кратчайшая кривая, содержащая все объекты внутри или на границе. Формально его можно определить его как границу выпуклой оболочки объединения всех объектов.

По заданному набору объектов найдите длину их выпуклого контура.

### Input

Первая строка входа содержит целое число  $n$  ( $1 \leq n \leq 20$ ) — количество объектов. Последующая строка состоит из  $n$  символов, задающих последовательность объектов слева направо. Каждый символ — одна из букв “S”, “C” или “T”, задающих квадрат, круг или треугольник, соответственно.

### Output

Выведите одно вещественное число — длину заданного выпуклого контура с абсолютной или относительной точностью не хуже  $10^{-6}$ .

### Example

standard input	standard output
4 TSTC	9.088434417
3 SCT	7.50914177324

## Problem E. Easy Equation

Input file:            *standard input*  
Output file:          *standard output*  
Time limit:           1 second  
Memory limit:        512 mebibytes

Задано целое число  $k$ , большее 1; можно доказать, что существует бесконечное количество троек целых положительных чисел  $(a, b, c)$ , удовлетворяющих следующему равенству:

$$a^2 + b^2 + c^2 = k(ab + bc + ca) + 1.$$

По заданным целым положительным числам  $n$  и  $k$  найдите  $n$  троек чисел  $(a_1, b_1, c_1)$ ,  $(a_2, b_2, c_2)$ ,  $\dots$ ,  $(a_n, b_n, c_n)$ , удовлетворяющих вышеприведённому равенству, причём все  $3n$  чисел  $a_1, \dots, a_n, b_1, \dots, b_n, c_1, \dots, c_n$  должны быть попарно различными целыми положительными числами, строго меньшими  $10^{100}$ .

### Input

Первая строка содержит два целых числа  $k$  и  $n$  ( $2 \leq k \leq 1000, 1 \leq n \leq 1000$ ) — константу  $k$  в равенстве и требуемое количество троек.

### Output

Выведите  $n$  строк.  $i$ -я из них должна содержать три целых положительных числа  $a_i, b_i$  и  $c_i$  —  $i$ -е из найденных Вами решений, удовлетворяющих требованиям задачи.

### Examples

standard input	standard output
2 8	1 2 6 3 10 24 12 35 88 15 28 84 4 5 18 14 33 90 40 104 273 21 60 152
3 3	1 3 12 8 21 87 44 165 615

## Problem F. Free Figurines

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 512 mebibytes

Матрёшки представляют собой набор деревянных кукол различного размера, последовательно вложенных одна в другую так, что меньшая кукла может быть вложена в большую, при этом непосредственно внутри одной матрёшки находится не более одной матрёшки.

Задан набор из  $n$  попарно различных матрёшек, обозначенных последовательными целыми числами от 1 до  $n$  по возрастанию размера. Если матрёшка  $a$  находится непосредственно внутри матрёшки  $b$ , будем называть матрёшку  $b$  *родительской* для матрёшки  $a$ ; если для матрёшки  $a$  не существует родительской матрёшки, то такая матрёшка называется *свободной*. Конфигурация всего набора матрёшек задаётся указанием родительской матрёшки для каждой матрёшки.

За один шаг разрешено делать следующие действия:

- Поместить свободную матрёшку внутрь другой свободной матрёшки, если та пуста.
- Открыть непустую свободную матрёшку и извлечь матрёшку, которая расположена непосредственно внутри.

Найдите наименьшее количество шагов, требуемое для того, чтобы получить из одной заданной конфигурации другую.

### Input

Первая строка входа содержит одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество матрёшек.

Вторая строка содержит последовательность из  $n$  целых чисел  $p_1, p_2, \dots, p_n$  ( $0 \leq p_k \leq n$ ), задающих начальную конфигурацию. Число  $p_k$  равно 0, если матрёшка  $k$  является свободной, или номер родительской матрёшки для матрёшки  $k$ .

Третья строка содержит последовательность из  $n$  целых чисел  $q_1, q_2, \dots, q_n$  ( $0 \leq q_k \leq n$ ), задающих конечную конфигурацию в том же самом формате.

Гарантируется, что обе конфигурации корректны: любая матрёшка всегда имеет размер меньше, чем родительская, и ни у каких двух матрёшек родительские матрёшки не совпадают.

### Output

Выведите одно целое число — наименьшее количество шагов, требуемое для того, чтобы перевести начальную конфигурацию в конечную.

### Examples

standard input	standard output
7 3 5 4 0 7 0 0 3 5 0 6 7 0 0	2
6 2 5 4 0 0 0 2 6 4 5 0 0	3

## Problem H. Hangar Hurdles

Input file: *standard input*  
 Output file: *standard output*  
 Time limit: 8 seconds  
 Memory limit: 512 mebibytes

План большого ангара, в котором будет размещён цех сборки самолётов, представляет собой квадрат  $n \times n$ , каждая единичная клетка которого может быть свободна или занята колонной, поддерживающей крышу ангара. Строки квадрата занумерованы последовательными целыми числами от 1 до  $n$  сверху вниз, столбцы занумерованы последовательными целыми числами от 1 до  $n$  слева направо.

Требуется, чтобы платформы, перевозящие комплектующие, могли свободно перемещаться между различными точками внутри ангара. Платформы имеют вид квадрата с нечётной стороной  $k$ , центр которого совпадает с центром единичной клетки. Платформа может двигаться вверх, вниз, вправо или влево, если она после сдвига целиком располагается в ангаре и не содержит ни одной клетки, занятой колонной.

Вам задано  $q$  пар клеток  $A_k$  и  $B_k$ . Для каждой пары клеток найдите размер наибольшей платформы с центром в  $A_k$ , которая может быть перемещена так, чтобы её центр был в  $B_k$ .

### Input

Первая строка входа содержит одно целое число  $n$  ( $2 \leq n \leq 1000$ ) — сторону ангара. Каждая из последующих  $n$  строк состоит ровно из  $n$  символов, задающих одну строку плана ангара. Символ “#” задаёт клетку с колонной, а символ “.” задаёт пустую клетку.

Следующая строка содержит целое число  $q$  ( $1 \leq q \leq 300\,000$ ) — количество запросов.  $k$ -я из этих строк содержит четыре целых числа  $r_{A_k}, c_{A_k}, r_{B_k}, c_{B_k}$  ( $1 \leq r_{A_k}, c_{A_k}, r_{B_k}, c_{B_k} \leq n$ ) — номер строки и столбца клеток  $A_k$  и  $B_k$  соответственно. Клетки  $A_k$  и  $B_k$  различны, кроме того, гарантируется, что обе эти клетки не заняты колоннами.

### Output

Выведите  $q$  строк.  $k$ -я из этих строк должна содержать одно целое число  $s_k$  — размер наибольшей платформы, с центром в  $A_k$ , которая может быть перемещена так, чтобы её центр оказался в  $B_k$ . Если ни для какого размера платформы это невозможно сделать,  $s_k$  должно быть равно 0.

### Example

standard input	standard output
7	1
.....#.	0
...#.#.	3
....#..	1
....###	1
....#..	
#.....	
.....	
5	
2 5 5 2	
2 5 3 6	
2 2 6 3	
2 2 6 6	
1 1 7 7	

## Problem K. Key Knocking

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            1 second  
Memory limit:         512 mebibytes

Горан экспериментирует со смарт-картами, содержащими криптографические ключи. В этой задаче *ключ* — это битовая строка длины  $3n$ , где  $n$  — целое положительное число. Биты ключа индексированы последовательными целыми числами от 1 до  $3n$  слева направо. *Весом* ключа называется количество пар различных соседних бит, увеличенное на 1. Например, вес ключа “000” равен 1, а вес ключа “011010100” равен 7.

Горан выяснил, что, посылая слабые разряды электричества на контакты карты, он может выполнять следующее действие: выбрать два произвольных соседних бита в ключе и поменять их значения на противоположные. Например, одна операция может сменить ключ “000” на ключ “110”.

По данному ключу размера  $3n$  найдите любую последовательность из не более  $n$  операций, которые преобразуют заданный ключ в ключ весом как минимум  $2n$ . Гарантируется, что решение всегда существует.

### Input

Первая строка входа содержит строку, состоящую из цифр “0” и “1” — первоначальный ключ. Длина ключа равна  $3n$ , где  $n$  — целое положительное число, такое, что  $1 \leq n \leq 10^5$ .

### Output

Первая строка содержит целое число  $m$ , где  $0 \leq m \leq n$  — количество операций в Вашем решении.

Последующие строки должны содержать  $m$  целых чисел  $a_1, a_2, \dots, a_m$ , описывающих Ваше решение. Число  $a_k$  — индекс левого бита из двух, которые изменяются на  $k$ -м шаге.

Если первоначальный ключ уже имеет вес  $2n$  или более, можно вывести только одну строку, содержащую число 0.

### Examples

standard input	standard output
000000000	3 2 5 6
111001000111	2 3 9
010101	0

## Problem M. Mighty Monster

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

В игре «Heroes of the Fight and Rage» вы сражаетесь с опасным монстром. Монстр за один ход может делать одно из следующих действий:

- **Rake** (обозначается символом 'R');
- **Bite** (обозначается символом 'B');
- **Laser breath** (обозначается символом 'L').

Чтобы защититься от монстра, вы, в свою очередь, должны делать следующие ходы:

- **Slice** (обозначается символом 'S'), если монстр применил "Rake";
- **Kick** (обозначается символом 'K'), если монстр применил "Bite";
- **Shield** (обозначается символом 'H'), если монстр применил "Laser breath";

У монстра есть опасный комбодар: когда монстр последовательно выполняет три попарно различных атаки, они складываются в один комбодар. В этом случае вам необходимо отвечать контрударом Combo Breaker, обозначаемым символом 'C'. Один символ 'C' блокирует всю комбинацию из трёх атак. Любые последующие ходы монстра после атаки считаются отдельно или как часть другого комбодара, то есть один ход монстра не может быть частью более, чем одного комбодара.

Вам задана строка, задающая действия монстра. Выведите последовательность Ваших действий, которая позволит Вам защититься от всех его атак.

### Input

Входной файл состоит из непустой строки, содержащей не более  $10^6$  символов и состоящей только из букв 'R', 'B' и 'L', задающей действия монстра.

### Output

Выведите одну строку — последовательность Ваших действий для того, чтобы защититься от заданных действий монстра.

### Example

standard input	standard output
RRBBLLR	SSKKHHS
RLLLLRR	CHCS
RBLBR	CKS



## Problem N. Needle Navigator

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            2 seconds  
Memory limit:         512 mebibytes

API симулятора компаса, встроенного в современные мобильные телефоны, устроен следующим образом: пусть стрелка компаса указывает в некотором направлении (между 0 и 359 градусами, причём северное направление соответствует 0, а восточное — 90), тогда поворот анимированной стрелки задаётся количеством градусов, на которое она должна повернуться.

Например, если стрелка указывает на север, а на вход симулятору подаётся значение 90, стрелка повернётся по часовой стрелке (положительные числа задают поворот по часовой стрелке) и будет показывать на север, а если подаётся значение  $-45$ , то произойдёт поворот против часовой стрелки и в итоге стрелка будет показывать на северо-запад.

Ваша задача найти *кратчайший путь* (угол с наименьшим абсолютным значением) между текущим и заданным положением стрелки компаса.

### Input

Первая строка входа содержит одно целое число  $n_1$  ( $0 \leq n_1 \leq 359$ ) — текущий угол поворота стрелки. Вторая строка содержит целое число  $n_2$  ( $0 \leq n_2 \leq 359$ ) — положение, которое стрелка должна в итоге принять.

### Output

Выведите одно число — минимальное по абсолютной величине количество градусов, на которое должна повернуться стрелка для того, чтобы принять требуемое положение. Положительное число обозначает поворот по часовой стрелке, отрицательное — против часовой.

В случае, если введённые позиции диаметрально противоположны, движение будет осуществляться по часовой стрелке.

### Example

standard input	standard output
315 45	90
180 270	90
45 270	-135

## Problem O. Make More Money!

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

Вы изучили значения курсов акций одной из компаний и перенеслись в прошлое на некоторое количество дней со 100 долларами в кармане, чтобы немного подзаработать на продаже и покупке этих акций.

Вы можете покупать и продавать только целое число акций; на момент Вашего прибытия количество акций у Вас на руках равно нулю. Общее количество акций на рынке равно  $10^5$ , так что ни в какой момент времени у Вас на руках не может быть более  $10^5$  акций.

Считая, что дополнительных затрат на покупку и продажу акций нет, цены на бирже меняются однократно, а ваши действия не влияют на курс акций, выясните, какую максимальную сумму Вы сможете заработать за время Вашего путешествия в прошлое.

### Input

Первая строка входа содержит одно целое число  $d$  ( $1 \leq d \leq 365$ ), количество дней, на которое Вы переноситесь в прошлое. Далее следуют  $d$  строк,  $i$ -я из которых содержит целое число  $p_i$  ( $1 \leq p_i \leq 500$ ) — цена, по которой Вы можете покупать или продавать акции в  $i$ -й день с момента Вашего прибытия.

### Output

Выведите одно целое число — максимальную сумму, которая может оказаться у вас на руках по прошествии  $d$  дней. Отметим, что ответ может превосходить  $2^{32}$ .

### Example

standard input	standard output
6	650
100	
200	
100	
150	
125	
300	

## Problem P. Several Sticks

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

Заданы  $n$  палочек различной длины. Выясните, можно ли выбрать три из этих палочек так, чтобы построить треугольник ненулевой площади.

### Input

Первая строка входа содержит одно целое число  $N$  ( $3 \leq N \leq 2 \cdot 10^4$ ) — количество палочек, которые у Вас есть. Далее следует строка, содержащая  $N$  целых положительных чисел, каждое из которых меньше  $2^{60}$  — длины палочек.

### Output

В случае, если можно выбрать три палочки так, чтобы из них получился невырожденный треугольник, выведит “possible”. В противном случае выведите “impossible”.

### Example

standard input	standard output
3 1 1 1	possible
5 3 1 10 5 15	impossible

## Problem Q. Falling Stones

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

Дана двумерная сетка, содержащая камни, препятствия и свободные места. Под воздействием силы гравитации камни начинают падать вниз (вдоль столбцов) до тех пор, пока они не наткнутся на препятствие, границу сетки или другой камень, который уже остановился, препятствия же остаются неподвижными.

По заданному начальному расположению камней и препятствий восстановите итоговую картинку.

### Input

Первая строка входа содержит два целых числа  $r$  и  $c$  ( $1 \leq r, c \leq 100$ ) — количество строк и столбцов сетки. Каждая из последующих  $r$  строк содержит по  $c$  символов: 'o' для камня, 'x' для препятствия и '.' для свободного пространства.

### Output

Выведите в аналогичном формате состояние сетки после того, как все камни упали.

### Examples

standard input	standard output
4 4 oooo x... ..x. x.xx	o... x.o. ..xo xxxx
4 3 o.o o.o o.. ...	... o.. o.o o.o

## Problem R. Regular Strings

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

Определим *k*-регулярную строку следующим образом.

Строка *s* является *k*-регулярной, если длина  $|s|$  этой строки кратна *k*, и если при разрезании строки на  $|s|/k$  подстрок длины *k* каждая последующая подстрока является циклическим сдвигом предыдущей подстроки вправо (то есть получена из предыдущей подстроки переносом последнего символа в начало)

Например, строка “хуzzхууzzххуz” является 3-регулярной, так как она может быть разбита на подстроки “хуз”, “zху”, “уzx” и “хуз”, и каждая следующая подстрока является циклическим сдвигом предыдущей.

По заданной строке определите наименьшее *k*, для которого данная строка является *k*-регулярной.

### Input

Вход содержит одну строку *s* ( $1 \leq |s| \leq 100$ ), состоящую из строчных латинских букв.

### Output

Выведите одно целое число — наименьшее значение *k*, для которого заданная строка является *k*-регулярной.

### Examples

standard input	standard output
qqqqqq	1
cddccddccddc	2
amppz	5