Задача A. Основная теорема арифметики

Имя входного файла:	`arithm.in`
Имя выходного файла:	`arithm.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Основная теорема арифметики гласит, что любое натуральное число $x > 1$ может быть единственным образом представлено в виде произведения $x = p_1^{a_1} \cdot \ldots \cdot p_k^{a_k}$, где $p_1 < \ldots < p_k$ — простые числа, $a_1, \ldots, a_k$ — положительные целые числа.

Ваша задача — написать программу, находящую такое представление для заданного числа x. Напомним, что натуральное число называется *простым*, если оно больше единицы и делится только на единицу и на себя.

Формат входного файла

Входной файл содержит целое число x ($2 \leq x \leq 2 \cdot 10^9$).

Формат выходного файла

В выходной файл выведите искомое представление числа x. В качестве знака умножения выводите `*` («звездочка»). Сомножитель вида p^a выводите как `p^a`. Если $a = 1$, то выводите просто `p`. Отделяйте сомножители от знака умножения пробелом.

Примеры

arithm.in	arithm.out
12	2^2 * 3
16	2^4
36	2^2 * 3^2

Задача B. Игра с шариками

Имя входного файла:	`balls.in`
Имя выходного файла:	`balls.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Действие одной очень популярной компьютерной игры, которая часто бывает установлена на карманных компьютерах происходит на квадратном поле размером 11 на 11, разбитом на 121 маленький квадратик.

Изначально в каждом квадратике находится шарик одного из пяти цветов: красного (обозначается символом `R`), синего (`B`), зеленого (`G`), желтого (`Y`), фиолетового (`V`). Назовем *связной областью*, содержащий данный шарик, все шарики, до которых можно добраться из данного, двигаясь каждый раз на один квадратик по вертикали или горизонтали, не выходя за границы игрового поля и проходя только по шарикам того же цвета, что и данный.

При выборе некоторого шарика автоматически выбираются все шарики, лежащие в одной связной области с ним. Если эта связная область содержит хотя бы 2 шарика, то эти шарики исчезают и игроку начисляется $n \cdot (n-1)$ очков, где n — количество шариков в связной области.

Задано начальное расположение шариков. Необходимо для каждого цвета определить, какое максимальное количество очков можно набрать за первый ход, выбрав один шарик такого цвета.

Формат входного файла

Входной файл содержит 11 строк по 11 символов в каждой — описание игрового поля.

Формат выходного файла

Для каждого цвета шариков в выходной файл выведите максимальное количество очков, которое можно набрать, выбрав шарик этого цвета. Следуйте формату, приведенному в примере.

Примеры

<table>
<tr><th>balls.in</th><th>balls.out</th></tr>
<tr><td><pre>RRRRRBBBGGG
RRRRRBBBGGG
RRRRRBBBGGG
RRRRRBBBGGG
RRRRRBBBGGG
RRRRRBBBGGG
RRRRRBBBGGG
YYYYYYYYYYY
YYYYYYYYYYY
YYYYYYYYYYY
VRVRVBVBVGV</pre></td><td><pre>R: 1190
G: 420
B: 420
Y: 1056
V: 0</pre></td></tr>
</table>

Задача C. Две окружности

Имя входного файла:	`circles.in`
Имя выходного файла:	`circles.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Заданы две окружности. Необходимо найти количество точек с целыми координатами, которые лежат строго внутри обеих окружностей.

Формат входного файла

Входной файл содержит две строки, каждая из которых описывает одну из окружностей. Описание окружности состоит из двух целых чисел: x, y и r, соответственно координат ее центра и ее радиуса ($-10^9 \le x, y \le 10^9$, $1 \le r \le 10^3$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

circles.in	circles.out
0 0 1 0 0 3	1

Задача D. Диплом

Имя входного файла:	`degree.in`
Имя выходного файла:	`degree.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Сергей скоро заканчивает обучение в институте. Уже написана пояснительная записка к выпускной квалификационной работе и оформлены все документы. Теперь Сергей задумался, не может ли он получить по результатам обучения диплом с отличием. В официальных документах он нашел следующее: «Студент получает диплом с отличием, если количество оценок «хорошо» не превышает 25% (в исключительных случаев допускается одна оценка «удовлетворительно»)». Более формально, студент получает диплом с отличием, если у него не более одной оценки «удовлетворительно», а количество оценок «хорошо» и «удовлетворительно» не превышает 25% от общего количества оценок.

Изучив свою зачетную книжку, Сергей составил список всех оценок, которые он получил за годы обучения в институте. Список получился достаточно большим — оценок оказалось n штук. Напишите программу, которая по этому списку определит, какой диплом получит Сергей в соответствии с указанным выше правилом.

Формат входного файла

Первая строка входного файла содержит число n ($1 \le n \le 100000$). Вторая строка входного файла содержит n чисел — оценки Сергея. Числом «5» обозначается оценка «отлично», числом «4» — оценка «хорошо», числом «3» — оценка «удовлетворительно». Каждое число во второй строке равно либо 3, либо 4, либо 5.

Формат выходного файла

В выходной файл выведите `Degree with honors`, если Сергей получает диплом с отличием, и `Ordinary degree` иначе.

Примеры

degree.in	degree.out
5 5 5 5 5 4	Degree with honors
4 3 3 4 5	Ordinary degree

Задача E. Текстовый редактор

Имя входного файла:	`editor.in`
Имя выходного файла:	`editor.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Андрей занимается разработкой нового текстового редактора. Первая работающая версия этого редактора поддерживает только однострочные документы. Этот недостаток будет устранен в следующей версии.

Основной возможностью этого редактора является возможность записи действий пользователя. Пользователь может совершить следующие действия:

- ввести очередную букву — в этом случае в список команд добавляется введенная буква;
- стереть последний набранный символ — в этом случае в список команд добавляется `#`;
- стереть всю строку — в этом случае в список команд добавляется `@`.

Если перед стиранием последнего введенного символа или всей строки строка пуста, то ничего не происходит.

Одна из основных проблем теперь — написать программу, которая будет воспроизводить действия пользователя по записанной последовательности команд. Ваша задача — написать такую программу. Она должна по списку команд находить строку, которая получается после их выполнения.

Формат входного файла

Входной файл содержит одну строку, в которой записан список команд. Его длина не превосходит 255. Каждый символ строки является либо строчной буквой латинского алфавита, либо символом `@`, либо символом `#`.

Формат выходного файла

В выходной файл выведите строку, получающуюся в результате выполнения всех команд из входного файла.

Примеры

editor.in	editor.out
`hellov#world@thisisasample`	`thisisasample`
`abc####hello`	`hello`

Задача F. Квадратный корень

Имя входного файла:	`sqroot.in`
Имя выходного файла:	`sqroot.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Введем в рассмотрение так называемые *0-1 матрицы* размером 4 на 4. Такая матрица — это квадратная таблица, содержащая 16 чисел $a_{i,j}$ ($i = 1 \ldots 4$, $j = 1 \ldots 4$), каждое из которых равно 0 или 1.

Произведением двух матриц A и B называется матрица $A \cdot B = C$, элементы которой вычисляются по формуле $c_{i,j} = (\sum_{k=1}^{4} a_{i,k} \cdot b_{k,j}) \bmod 2$. *Квадратным корнем* из матрицы A называется 0-1 матрица B, такая что $B \cdot B = A$.

Задана некоторая 0-1 матрица размера 4 на 4. Вычислите ее квадратный корень или установите, что его не существует.

Формат входного файла

Входной файл содержит четыре строки, каждая из которых содержит четыре числа (каждое из этих чисел — либо 0, либо 1). j-ое число i-ой строки соответствует элементу $a_{i,j}$ заданной матрицы A.

Формат выходного файла

Выведите в выходной файл квадратный корень из заданной матрицы в формате, аналогичном входному файлу. Если квадратного корня не существует — выведите в выходной файл `NO SOLUTION`. Если решений несколько, выведите любое.

Примеры

sqroot.in	sqroot.out
1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 0	NO SOLUTION

Задача G. Тор

Имя входного файла:	`torus.in`
Имя выходного файла:	`torus.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Представьте себе прямоугольник размером n клеток по вертикали на m клеток по горизонтали. После этого мысленно «склейте» его верхнюю сторону с нижней, а левую — с правой. В результате получится тор.

Назовем две клетки *соседними*, если у них имеется общая сторона. Пусть за одну секунду можно из клетки перейти в любую соседнюю с ней. За какое минимальное количество секунд можно из клетки (r_1, c_1) попасть в клетку (r_2, c_2)? Первое число в обозначении клетки есть номер строки прямоугольника, в которой она находится, второе — номер столбца.

Формат входного файла

Входной файл содержит шесть целых чисел: n, m, r_1, c_1, r_2, c_2 ($2 \le n, m \le 10^9$, $1 \le r_1, r_2 \le n$, $1 \le c_1, c_2 \le m$).

Формат выходного файла

Выведите в выходной файл ответ на задачу.

Примеры

torus.in	torus.out
10 10 5 5 1 1	8
10 10 9 9 1 1	4

Задача H. Два треугольника

Имя входного файла:	`twotri.in`
Имя выходного файла:	`twotri.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Заданы два треугольника: ABC и DEF. Необходимо определить, являются ли они подобными.

Формат входного файла

Первая строка входного файла содержит координаты вершин треугольника ABC: x_A, y_A, x_B, y_B, x_C, y_C. Вторая строка входного файла содержит в аналогичном формате координаты вершин второго треугольника.

Все числа во входном файле целые и не превосходят 5000 по абсолютному значению. Оба описанных во входном файле треугольника имеют ненулевую площадь.

Формат выходного файла

Выведите в выходной файл слово `YES`, если заданные во входном файле треугольники подобны, и слово `NO` — в противном случае.

Примеры

twotri.in	twotri.out
0 0 1 0 0 1 10 10 12 10 10 12	YES
0 0 1 0 0 1 0 0 10 0 0 5	NO