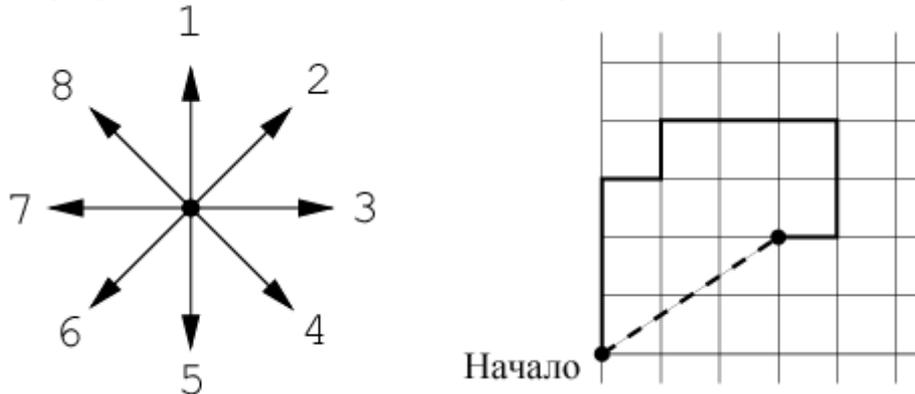


## Задача 1

Найти закопанный пиратами клад просто: всё, что для этого нужно – это карта. Как известно, пираты обычно рисуют карты от руки и описывают алгоритм нахождения клада так: «Встаньте около одинокой пальмы. Пройдите тридцать шагов в сторону леса, потом семнадцать шагов в сторону озера, ..., наконец десять шагов в сторону большого бульжника. Клад находится под ним». Большая часть таких указаний просто сводится к прохождению какого-то количества шагов в одном из восьми направлений (1 – север, 2 – северо-восток, 3 – восток, 4 – юго-восток, 5 – юг, 6 – юго-запад, 7 – запад, 8 – северо-запад) (см. рис). Длина шага в любом направлении равна 1.

Путешествие по такому пути обычно является прекрасным способом посмотреть окрестности, однако в наше время постоянной спешки ни у кого нет времени на это. Поэтому кладоискатели хотят идти напрямую в точку, где зарыт клад. Например, вместо того, чтобы проходить три шага на север, один шаг на восток, один шаг на север, три шага на восток, два шага на юг и один шаг на запад, можно пройти напрямую, используя около 3.6 шага (см. рис).



Вам необходимо написать программу, которая по указаниям пиратов определяет точку, где зарыт клад.

### Формат входных данных

Первая строка содержит число  $N$  – число указаний ( $1 \leq N \leq 40$ ). Последующие  $N$  строк содержат сами указания – номер направления (целое число от 1 до 8) и количество шагов (целое число от 1 до 1000). Числа разделены пробелами.

### Формат выходных данных

Выведите координаты  $X$  и  $Y$  точки (два вещественных числа, разделённые пробелом), где зарыт клад, считая, что ось  $Ox$  направлена на восток, а ось  $Oy$  – на север. В начале кладоискатель должен стоять в начале координат. Координаты необходимо вывести с погрешностью не более  $10^{-3}$ .

### Примеры

| Вход  | Выход        |
|---|--------------|
| 6<br>1 3<br>3 1<br>1 1<br>3 3<br>5 2<br>7 1 | 3.000 2.000  |
| 1<br>8 10                                   | -7.071 7.071 |

## Задача 2

Многоугольник (не обязательно выпуклый) на плоскости задан координатами своих вершин. Требуется подсчитать количество точек с целочисленными координатами, лежащих внутри него (но не на его границе).

### Формат входных данных

В первой строке содержится  $N$  ( $3 \leq N \leq 1000$ ) – число вершин многоугольника. В последующих  $N$  строках идут координаты  $(X_i, Y_i)$  вершин многоугольника в порядке обхода по часовой стрелке.  $X_i$  и  $Y_i$  - целые числа, по модулю не превосходящие 1000000.

### Формат выходных данных

Вывести одно число – искомое число точек.

### Примеры

| Вход                              | Выход |
|-----------------------------------|-------|
| 4<br>-1 -1<br>-1 1<br>1 1<br>1 -1 | 1     |
| 3<br>0 0<br>0 2<br>2 0            | 0     |

## Задача 3

Для того чтобы проверить, как её ученики умеют считать, Мария Ивановна каждый год задаёт им на дом одну и ту же задачу – для заданного натурального  $A$  найти минимальное натуральное  $N$  такое, что  $N$  в степени  $N$  ( $N$ , умноженное на себя  $N$  раз) делится на  $A$ . От года к году и от ученика к ученику меняется только число  $A$ .

Вы решили помочь будущим поколениям. Для этого вам необходимо написать программу, решающую эту задачу.

### Формат входных данных

На вход подаётся единственное число  $A$  ( $1 \leq A \leq 1000000000$  – на всякий случай; вдруг Мария Ивановна задаст большое число, чтобы «завалить» кого-нибудь...).

### Формат выходных данных

Вывести единственное число  $N$ .

### Примеры

| Вход | Выход |
|------|-------|
| 8    | 4     |
| 13   | 13    |

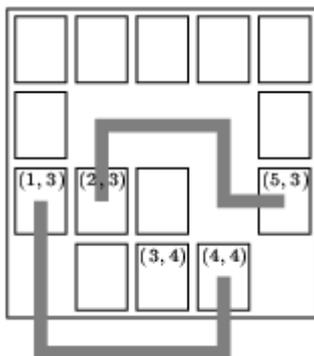
## Задача 4

Вы являетесь одним из разработчиков новой компьютерной игры. Игра происходит на прямоугольной доске, состоящей из  $W \times H$  клеток. Каждая клетка может либо содержать, либо не содержать фишку (см. рисунок).

Важной частью игры является проверка того, соединены ли две фишки путем, удовлетворяющим следующим свойствам:

1. Путь должен состоять из отрезков вертикальных и горизонтальных прямых.
2. Путь не должен пересекать других фишек.

При этом часть пути *может* оказаться вне доски. Например:



Фишки с координатами (1,3) и (4,4) могут быть соединены. Фишки с координатами (2,3) и (5,3) тоже могут быть соединены. А вот фишки с координатами (2,3) и (3,4) соединить нельзя – любой соединяющий их путь пересекает другие фишки.

Вам необходимо написать программу, проверяющую, можно ли соединить две фишки путем, обладающим вышеуказанными свойствами, и, в случае положительного ответа, определяющую минимальную длину такого пути (считается, что путь имеет изломы, начало и конец только в центрах клеток (или «мнимых клеток», расположенных вне доски), а отрезок, соединяющий центры двух соседних клеток, имеет длину 1).

### Формат входных данных

Первая строка содержит два натуральных числа:  $W$  – ширина доски,  $H$  – высота доски ( $1 \leq W, H \leq 75$ ). Следующие  $H$  строк содержат описание доски: каждая строка состоит ровно из  $W$  символов: символ «X» (заглавная английская буква «экс») обозначает фишку, символ «.» (точка) обозначает пустое место. Все остальные строки содержат описания запросов: каждый запрос состоит из четырёх натуральных чисел, разделённых пробелами –  $X_1, Y_1, X_2, Y_2$ , причём  $1 \leq X_1, X_2 \leq W, 1 \leq Y_1, Y_2 \leq H$ . Здесь  $(X_1, Y_1)$  и  $(X_2, Y_2)$  – координаты фишек, которые требуется соединить (левая верхняя клетка имеет координаты (1,1)). Гарантируется, что эти координаты не будут совпадать (кроме последнего запроса; см. далее). Последняя строка содержит запрос, состоящий из четырёх чисел 0; этот запрос обрабатывать не надо. Количество запросов не превосходит 20.

### Формат выходных данных

Для каждого запроса необходимо вывести одно целое число на отдельной строке – длину кратчайшего пути, или 0, если такого пути не существует.

### Примеры

| Вход   | Выход                                  |
|--|--|
| <pre> 5 4 XXXXX X...X XXX.X .XXX. 2 3 5 3 1 3 4 4 2 3 3 4 0 0 0 0                     </pre> | <pre> 5 6 0                     </pre> |
| <pre> 4 4 XXXX .X.. ..X. X... 1 1 3 1 0 0 0 0                     </pre>                     | <pre> 4                     </pre>     |

## Задача 5

Во время недавних раскопок на Марсе были обнаружены листы бумаги с таинственными символами на них. После долгих исследований учёные пришли к выводу, что надписи на них на самом деле могли быть обычными числовыми равенствами. Если бы этот вывод оказался верным, это доказало бы не только то, что на Марсе много лет назад были разумные существа, но и то, что они уже умели считать...

Ученые смогли понять, что в этом случае означают найденные символы, и перевели эти равенства на обычный язык – язык цифр, скобок, знаков арифметических действий и равенства. Кроме того, из других источников было получено веское доказательство того, что марсиане знали только три операции – сложение, умножение и вычитание (марсиане никогда не использовали «унарный минус»: вместо «-5» они писали «0-5»). Также ученые доказали, что марсиане не наделяли операции разным приоритетом, а просто вычисляли выражения (если в них не было скобок) слева направо: например,  $3+3*5$  у них равнялось 30, а не 18.

К сожалению, символы арифметических действий марсиане почему-то наносили специальными чернилами, которые, как оказалось, были не очень стойкими, и поэтому в найденных листках между числами вместо знаков действий были пробелы. Если вся вышеизложенная теория верна, то вместо этих пробелов можно поставить знаки сложения, вычитания и умножения так, чтобы равенства стали верными. Например, если был найден лист бумаги с надписью «18=7 (5 3) 2», то возможна такая расстановка знаков: « $18=7+(5-3)*2$ » (помните про то, в каком порядке марсиане вычисляют выражения!). В то же время, если попался лист с надписью «5=3 3», то марсиане явно не имели в виду числового равенства, когда писали это...

Вы должны написать программу, находящую требуемую расстановку знаков или сообщаящую, что таковой не существует.

### Формат входных данных

Первая строка состоит из натурального (целого положительного) числа, не превосходящего  $2^{30}$ , знака равенства, и последовательности натуральных чисел (не более десяти), произведение которых также не превосходит  $2^{30}$ . Некоторые группы чисел (одно или более) могут быть окружены скобками. Длина входной строки не будет превосходить 80 символов, и других ограничений на количество и вложенность скобок нет. Между двумя соседними числами, не разделенными скобками, всегда будет хотя бы один пробел, во всех остальных местах может быть любое (в том числе и 0) число пробелов (естественно, внутри числа пробелов нет).

### Формат выходных данных

Необходимо вывести одну строку, содержащую полученное равенство (т.е., исходное равенство со вставленными знаками арифметических действий). В случае если требуемая расстановка знаков невозможна, вывести строку, состоящую из единственного числа «-1». Выходная строка не должна содержать пробелов.

### Примеры

| Вход         | Выход        |
|--------------|--------------|
| 18=7 (5 3) 2 | 18=7+(5-3)*2 |
| 5= 3 3       | -1           |