

Задача А. Максимальное произведение

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан массив a из n целых чисел. Найдите пару индексов i, j такую, что $i < j$ и $(j - i) \cdot \min(a_i, a_j)$ максимально. В качестве ответа выведите значение этого произведения.

Формат входных данных

Первая строка содержит целое число n ($2 \leq n \leq 5 \cdot 10^5$). Вторая строка содержит n целых чисел a_i ($1 \leq a_i \leq 10^5$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
4 1 6 4 3	6

Задача В. Квадратами, квадратами

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У вас есть матрица a из n строк и m столбцов, состоящая из нулей, а также матрица b из n строк и m столбцов, состоящая из единиц и двоек. Вы хотите превратить матрицу a в матрицу b .

Для этого вы можете выбрать некоторое k , а дальше неограниченное число раз превратить любой квадрат матрицы со стороной k в единицы или двойки. Квадрат должен лежать целиком внутри матрицы. Понятно, что при $k = 1$ это всегда возможно. Определите, для какого наибольшего k этого можно достичь?

Формат входных данных

В первой строке даны два целых числа n и m ($1 \leq n, m \leq 100$). Следующие n строк содержат по m символов: единицы и двойки — матрицу b .

Формат выходных данных

Выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
3 6 121222 121222 121222	3

Замечание

В примере можно сначала первые три колонки заполнить единицами, затем колонки 2 — 4 заполнить двойками, затем колонки 3 — 5 заполнить единицами, и, наконец, колонки 4 — 6 двойками.

Задача С. Покраска матрицы

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дана белая матрица, состоящая из n строк и m столбцов. В i -й строке вы должны покрасить либо первые k_i ячеек (то есть левые), либо последние $m - k_i$ ячеек (правые). Пусть теперь в столбце t_j ячеек покрашено. Красотой столбца назовем величину $t_j \cdot (n - t_j)$. Красотой матрицы назовем сумму красот всех столбцов. Определите минимальную возможную красоту матрицы.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^4$). Следующие n строк содержат по одному целому числу k_i ($0 \leq k_i \leq m$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
2 3 2 1	1
4 4 1 1 4 1	3

Замечание

Во втором примере можно покрасить, например, так:

```
#...  
#...  
....  
#...
```

Это приведет к красоте $3 \cdot 1 + 0 \cdot 4 + 0 \cdot 4 + 0 \cdot 4 = 3$.

Задача D. Подстрока зеркального отражения

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У вас есть строка S длины n , состоящая из строчных букв английского алфавита. Вы можете k раз выполнить следующую операцию.

Обозначим за S^R развернутую строку S (записанную справа налево). Посмотрим на строку SS^R (конкатенацию строк S и S^R) и выберем любую подстроку длины n этой строки в качестве новой строки S .

Какую лексикографически минимальную строку можно получить такими операциями?

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq n \leq 5000$, $1 \leq k \leq 10^9$). Вторая строка содержит строку S , состоящую из n строчных букв английского алфавита.

Формат выходных данных

Выведите одну строку — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
5 1 badba	aabda

Замечание

В примере есть строка $badba$, тогда SS^R выглядит как $badbaabdab$, лексикографически минимальная подстрока длины 5 это $aabda$.

Задача Е. Строка с наименьшим хешем

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Давно был придуман хеш для строки

```
function hash(s):  
    answer = 0  
    for each valid index i into s, starting from 0:  
        answer = (answer * 127 + ord(s[i])) mod 1000000000000037  
    return answer
```

Возвращаемое значение называется хешем строки s . В псевдокоде $\text{ord}(s[i])$ — это ASCII-код i -го символа строки s .

У каждого из трех сокомандников есть любимая строка длиной n : у Дениса a , у Тихона b , а у Севы c . Теперь они решили создать новую строку s с той же длиной n , так что для каждой позиции i мы имеем $s_i = a_i$ или $s_i = b_i$ или $s_i = c_i$.

Среди всех таких строк Денис, Тихон и Сева хотят найти строку s с наименьшим хешем. Вам дана длина строк n и три любимых строки a , b и c . Найдите значение наименьшего возможного хеша строки s с указанным выше свойством.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 28$). Следующие три строки содержат три строки длины n , состоящих из заглавных букв английского алфавита.

Формат выходных данных

Выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
4 ELLY KRIS STAN	142572564

Замечание

В первом примере подходит строка ELAN.

Задача F. Бегаем между лекциями

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Лекции проходят в двух корпусах, назовем их корпус 1 и корпус 2. Всего сегодня состоится n лекций. Лекция i начнется в момент времени s_i в корпусе p_i . Преподаватель каждой лекции чуть-чуть подождет опаздывающих, так что если вы придете в корпус в момент s_i или раньше, то успеете. Лекция продлится чуть меньше, чем одну единицу времени, поэтому ровно в момент $s_i + 1$ вы уже будете свободны.

Вы хотите посетить некоторые лекции. День начинается (и вы просыпаетесь) в момент времени 0, а проснуться вы можете в любом корпусе на ваш выбор. Вы можете переходить из корпуса 1 в корпус 2 и обратно. Время, необходимое для перехода между двумя корпусами, составляет $d + k \times j$, где j — количество лекций, на которых вы были в этот день к моменту начала перехода.

Определите, на каком наибольшем числе лекций вы сможете побывать.

Формат входных данных

Первая строка содержит три целых числа n , d и k ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq d \leq 10^{12}$, $0 \leq k \leq 10^{12}$). Следующие n строк содержат по два целых числа p_i , s_i ($1 \leq p_i \leq 2$, $1 \leq s_i \leq 10^{12}$, $s_i \neq s_j$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 3 0 1 1 1 2 1 10 2 5 2 6	4
5 3 1 1 1 1 2 1 10 2 5 2 6	3

Замечание

В первом примере вы можете проснуться в корпусе 1, посетить первую и вторую лекции, затем с момента 3 уйти в корпус 2 (и прийти туда в момент 6), затем послушать пятую лекцию, в момент 7 перейти в корпус 1, прийти в момент 10 и посетить третью лекцию. Итого 4 лекции.

Задача G. Лабиринт с потайными дверьми

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Это интерактивная задача

Вы заперты в лабиринте, в котором нет выхода, и хотите нарисовать его карту. В лабиринте могут быть до двух скрытых потайных дверей. Попад в потайную дверь, вы падаете через неё и оказываетесь в другой части лабиринта. Чтобы усложнить ситуацию, в лабиринте мало отличительных признаков, поэтому у вас нет простого способа распознать ранее посещенные места. Тем не менее, у вас есть острое чувство направления, поэтому вы всегда можете определить, где север.

Лабиринт представляет собой прямоугольную сетку, где каждая ячейка является либо стеной, либо открытым пространством, либо потайной дверью. Из открытого пространства вы можете различить только то, какие из четырёх соседних ячеек (на севере, востоке, юге и западе) являются стенами, и затем можете перейти в любую соседнюю ячейку, которая не является стеной. Если вы попадаете в потайную дверь, вы падаете в неё, но у вас есть время понять, являются ли стенами ячейки, соседствующие с потайной дверью.

Известно, что в лабиринте можно из любого места добраться в любое другое место (но это может потребовать использования потайной двери, как в примере ниже). Лабиринт также состоит из одной области — если бы потайные двери были заменены на простые открытые пространства, всё равно было бы возможно добраться из любого места в любое другое место. Концы потайных дверей могут находиться в любом открытом пространстве лабиринта, но не на другой потайной двери, и концы двух потайных дверей не находятся в одной позиции. Ваша стартовая точка не является ни потайной дверью, ни концом потайной двери.

Постройте полную карту лабиринта.

Протокол взаимодействия

Когда вы оказываетесь в клетке, вы получаете информацию о её окружении, в том числе в самом начале взаимодействия. Информация о вашем текущем окружении представлена в одной строке. Строка начинается с четырёх символов cN , cE , cS и cW , описывающих окружение ячейки, в которую вы только что переместились (или, для первого раунда, начальной ячейки), в направлениях север, восток, юг и запад соответственно. Каждый символ либо решётка ($\#$), указывающий на то, что ячейка в этом направлении является стеной, либо точка (\cdot), указывающий на то, что вы можете пройти туда. Затем в той же строке следует либо слово `trap`, если ячейка, в которую вы переместились, является потайной дверью, либо `ok`, если это не так. Если ячейка является потайной дверью, строка содержит третью строку, описывающую окружение конечной точки потайной двери в том же формате.

Существует два типа действий: перемещение в соседнюю ячейку и сообщение о том, что вы построили план лабиринта. Чтобы переместиться в соседнюю ячейку, выведите одну строку, содержащую один из символов N , E , S и W , соответствующий перемещению на север, восток, юг и запад соответственно. Чтобы сообщить о том, что вы завершили исследование и хотите вывести ответ, выведите строку с текстом `done`, за которой выведите полную карту лабиринта.

Выведите карту в виде прямоугольной сетки минимального размера, включающую все увиденные стены. Сначала выведите строку с двумя целыми числами h и w , указывающими высоту и ширину сетки. Затем выведите h строк (с севера на юг), каждая из которых содержит ровно w символов (с запада на восток), используя следующие символы:

- $\#$ — для стен и для других недоступных ячеек, находящихся вне лабиринта.
- S — для вашей стартовой позиции.
- A и B — для местоположений двух потайных дверей.
- a и b — для соответствующих местоположений конечных точек двух потайных дверей.
- Точку (\cdot) — для остальных проходимых ячеек.

Не имеет значения, какую из потайных дверей вы обозначите как A , а какую как B , пока a — это конечная точка потайной двери, обозначенной как A , и аналогично для b . Если существует только одна потайная дверь, вы можете обозначить её либо как A , либо как B . После вывода карты взаимодействие заканчивается, и ваша программа должна завершиться.

Гарантируется, что в лабиринте не более 500 достижимых позиций. Ваше решение может совершить не более 10^5 шагов. Пытаться проходить в стену запрещено. Не забудьте очистить буфер стандартного вывода после каждого действия.

Примеры

стандартный ввод	стандартный вывод
.##. ok	W
..## ok	N
#..# ok	E
##.. ok	done 4 4 #### #..# #S# ####
##. ok	E
##. trap .###	N
##.. trap ##.	E
##. ok	E
##. ok	E
##. trap .###	done 4 7 ##### #b.SAB# #####a# #####

Задача I. Криптография и панграммы

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На ФПМИ нам нравится отправлять друг другу панграммы — это фразы, которые содержат каждую букву английского алфавита хотя бы раз. Один из распространенных примеров панграммы — THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. На парах по криптографии мы быстро узнали, что очень трудно разложить на множители произведения двух больших простых чисел, поэтому мы разработали схему шифрования, основанную на этом факте. Сначала мы сделали некоторые приготовления:

- Мы выбрали 26 различных простых чисел, ни одно из которых не больше некоторого целого числа n .
- Мы отсортировали эти простые числа в порядке возрастания. Затем мы присвоили наименьшее простое число букве A, второе по величине простое число — букве B и так далее.
- Каждый член жюри запомнил этот список.

Теперь, когда мы хотим отправить панграмму в качестве сообщения, мы сначала убираем все пробелы, чтобы сформировать исходное сообщение. Затем мы записываем произведение простого числа для первой буквы исходного текста и простого числа для второй буквы исходного текста. Затем мы записываем произведение простых чисел для второй и третьей букв исходного текста и так далее, заканчивая произведением простых чисел для предпоследней и последней букв исходного текста. Этот новый список значений является нашим шифром. Количество значений на одно меньше, чем количество символов в исходном сообщении.

Например, предположим, что $n = 103$ и мы решили использовать первые 26 нечетных простых чисел, потому что беспокоимся, что разложить четные числа на множители слишком легко. Тогда $A = 3, B = 5, C = 7, D = 11$ и так далее, вплоть до $Z = 103$. Также предположим, что мы хотим зашифровать вышеупомянутую панграмму THE QUICK..., так что наше исходное сообщение — THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG. Тогда первое значение в нашем шифре — это 73 (простое для T) умноженное на 23 (простое для H) = 1679; следующее значение — 299 и так далее, заканчивая 1007.

Мы дадим вам сообщение шифра и значение n , которое мы использовали. Мы не скажем вам, какие простые числа мы использовали или как расшифровать шифр. Вы сможете восстановить исходное сообщение?

Формат входных данных

Первая строка содержит два целых числа n и k ($101 \leq n \leq 10^{100}$, $25 \leq k \leq 100$), где k это длина шифра. Следующая строка содержит k целых чисел — сам шифр.

Формат выходных данных

Выведите строку из $k + 1$ заглавной буквы английского алфавита — исходное сообщение.

Пример

стандартный ввод	стандартный вывод
103 34 1679 299 793 4819 2291 203 259 185 335 3551 4717 4183 799 901 5141 3007 2449 3397 2537 4189 3763 4399 1079 871 4891 1679 299 533 123 309 10403 1111 583 1007	THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG

Замечание

Переводы строки в примере даны только для удобства форматирования, ввод содержит две строки.

Задача J. Доставка треугольниками

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В точке $(0; 0)$ плоскости расположен склад. Имеется n заказов, каждый из них расположен в точке с координатами (x_i, y_i) . Курьер должен доставлять заказы парами, то есть он выбирает два заказа, едет по прямой в точку с первым заказом, затем оттуда по прямой в точку со вторым заказом, затем возвращается на склад.

Нужно распределить заказы на пары так, чтобы маршрут курьера был минимален, и при этом маршрут не имел самопересечений (кроме как на складе). Определите длину этого маршрута.

Формат входных данных

Первая строка содержит четное целое число n ($2 \leq n \leq 500$). Следующие n строк содержат по два целых числа x_i, y_i ($-10^6 \leq x_i, y_i \leq 10^6$). Никакие две точки не совпадают. Никакие две точки не лежат на одной прямой со складом.

Формат выходных данных

Выведите одно число — минимальную длину маршрута с абсолютной или относительной погрешностью не более 10^{-6} .

Пример

стандартный ввод	стандартный вывод
4 -1 1 -1 4 1 1 1 4	17.074638376

Замечание

В первом примере самый короткий путь, удовлетворяющий всем правилам, следующий: $(0, 0) \rightarrow (-1, 1) \rightarrow (-1, 4) \rightarrow (0, 0) \rightarrow (1, 4) \rightarrow (1, 1) \rightarrow (0, 0)$. Если бы не было правила про самопересечение, курьер мог бы сначала доставить заказы в первую и третью точки, затем во вторую и четвертую точки, но этот путь не разрешен, поскольку пути от склада до второй точки и от первой до третьей точек пересекаются.

Задача К. Качалка в двенашке

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Демид каждый день ходит в качалку в двенашке по следующей схеме. Если в некоторый день он хочет выжать лёжа a килограмм, и ему это удаётся, он добавляет к весу на следующий день 25 килограмм. Если ему не удаётся выжать a килограмм, он уменьшает вес на 10 процентов, округляет вниз до ближайшего веса, кратного 25 килограммам, и пробует на следующий день. Новый вес может быть равен 0 (не факт, что Демид с ним справится).

Нужно проверить, могла ли получиться ситуация, что k дней назад Демид пробовал выжать a килограмм, а сегодня пробовал выжать b килограмм.

Формат входных данных

Единственная строка содержит три целых числа a , b и k ($25 \leq a, b \leq 10^{18}$, $1 \leq k \leq 10^{18}$). Числа a и b кратны 25.

Формат выходных данных

Если такая ситуация могла быть, выведите Yes, иначе выведите No.

Примеры

стандартный ввод	стандартный вывод
100 100 4	Yes
100 150 3	No

Задача L. Подстрока с разным количеством

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана строка из символов A , B , C , нужно найти длину наибольшего непрерывного подотрезка этой строки, что символы в подотрезке встречаются разное количество раз (или не встречаются вовсе). Другими словами, все буквы, которые присутствуют на подотрезке, должны присутствовать разное число раз.

Формат входных данных

Дана одна строка из символов A , B , C , длина строки не превосходит 10^6 .

Формат выходных данных

Выведи одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
СВВААВСАС	6
ААААА	5

Задача М. Домножения и повороты

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана матрица a целых чисел, состоящая из n строк и m столбцов. Можно выполнять следующие операции:

- *rotateRow* i k — повернуть i -ю строку на k элементов вправо ($a'[i][j] = a[i][j - k]$)
- *rotateCol* j k — повернуть j -й столбец на k элементов вниз ($a'[i][j] = a[i - k][j]$)
- *negRow* i — домножить все числа в i -й строке на -1 , если ни одно из этих чисел не домножалось на -1 ранее
- *negCol* j — домножить все числа в j -м столбце на -1 , если ни одно из этих чисел не домножалось на -1 ранее

В первых двух операциях числа прокручиваются по кругу, смотри замечание для большего понимания. Используя не более чем $5nm$ операций, нужно максимизировать сумму чисел в матрице.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 70$). Следующие n строк содержат по m целых чисел a_{ij} ($-10^4 \leq a_{ij} \leq 10^4$).

Формат выходных данных

В первой строке выведите максимальную возможную сумму и t — число операций, необходимых для этого. Следующие t строк должны содержать операции в формате, описанном в условии, по одной операции в строке, в том порядке, в котором их нужно применять.

- При выполнении *rotateRow* i k должны выполняться условия $1 \leq i \leq n$, $1 \leq k < m$.
- При выполнении *rotateCol* j k должны выполняться условия $1 \leq j \leq m$, $1 \leq k < n$.
- При выполнении *negRow* i должно выполняться условие $1 \leq i \leq n$.
- При выполнении *negCol* j должно выполняться условие $1 \leq j \leq m$.

Пример

стандартный ввод	стандартный вывод
3 2	18 3
4 -2	rotateCol 1 2
1 0	rotateRow 2 1
-4 -7	negCol 2

Замечание

В примере матрица будет преобразовываться следующим образом

```
4 -2    1 -2    1 -2    1 2
1 0 -> -4 0 -> 0 -4 -> 0 4
-4 -7    4 -7    4 -7    4 7
```