

Для всех задач:

Входной файл:	<i>input.txt</i>
Выходной файл:	<i>output.txt</i>
Ограничение по памяти:	256 МБ
Максимальное количество баллов за задачу:	100 баллов

Задача 1. Максимальное расстояние

Ограничение по времени: 1 секунда на тест

На плоскости задано N точек. Найти расстояние между двумя наиболее удаленными друг от друга точками.

Входные данные

Первая строка входного файла содержит число N ($2 \leq N \leq 100000$). Далее в файле записано N пар целых чисел, задающих координаты точек. Все координаты по модулю не превышают 10^4 .

Выходные данные

В выходной файл необходимо вывести пару чисел — номера точек, для которых достигается максимум расстояния. Точки нумеруются по порядку, начиная с 1. Если решений несколько, то следует вывести любое из них.

Примеры

<i>input.txt</i>	<i>output.txt</i>
4 0 1 1 0 0 0 1 1	4 3

Задача 2. Диофантовы уравнения

Ограничение по времени: 1 секунда на тест

Диофантово уравнение имеет вид $A \cdot X + B \cdot Y = D$, где:

A, B, D — целые коэффициенты, X и Y — целые неизвестные.

Решение диофантова уравнения — это пара целых чисел $\langle X, Y \rangle$, при которых верно равенство в уравнении.

Вам необходимо найти решения заданных диофантовых уравнений.

Входные данные

В первой строке входного файла записано целое число N — количество уравнений ($0 < N \leq 100000$). В каждой из следующих строк записано по три целых числа A_i, B_i, D_i — коэффициенты уравнения ($0 < A_i, B_i, D_i \leq 10^9$).

Выходные данные

Каждая строка выходного файла должна содержать по два числа X_i и Y_i — решение i -го уравнения. Если решений несколько, то выдайте любое. Все числа в ваших ответах должны входить в 32-битовые знаковые целые числа. Если решений уравнения нет, выдайте произвольные два числа вместо решения.

Пример

<i>input.txt</i>	<i>output.txt</i>
5 1 1 5 2 3 1 50 70 10 2 2 1 50 71 10	0 5 -1 1 3 -2 666 666 57 -40

Задача 3. Деление длинных чисел

Ограничение по времени:

1 секунда на тест

Для заданной последовательности пар целых чисел выдать целую часть и остаток от деления первого числа каждой пары на второе.

Входные данные

В первой строке входного файла записано число N — количество пар чисел в последовательности ($1 \leq N \leq 100$). В следующих $2N$ строках записано по одному числу. Длина каждого числа не более 1000. Все числа десятичные длинные, неотрицательные.

Выходные данные

В выходной файл для каждой пары заданной последовательности чисел необходимо вывести два числа – целую часть и остаток от деления первого числа на второе, по одному на строке.

Пример

<i>input.txt</i>	<i>output.txt</i>
2	11
154	11
13	42626
100000999	403
2346	

Задача 4. Кокосы

Ограничение по времени:

1 секунда на тест

В газетах 1 апреля 2003 года сообщалось о кораблекрушении судна в Индийском океане. Пять человек и одна обезьянка спаслись на необитаемом острове. В первый день пребывания на острове они собирали кокосы. Среди ночи один из них проснулся и решил разделить кокосы. Он разделил кокосы на 5 равных частей. Однако остался 1 кокос, который он отдал обезьянке. Свою часть он взял с собой и пошел досыпать. Далее второй, третий, четвертый и пятый поступили аналогичным образом, и каждый раз оставался 1 кокос, который они с удовольствием отдавали обезьянке. Напрашивается вопрос: сколько было кокосов.

Однако мы предлагаем решить несколько другую задачу (обратную). Найдите максимально возможное число людей (обезьянка всегда одна), которые могут проделать рассмотренную ночную операцию, при условии, что известно количество собранных кокосов.

Входные данные

В первой строке целое число – количество собранных кокосов. Число собранных кокосов не превышает 5000000 штук.

Выходные данные

В выходной файл необходимо вывести одно целое число — максимальное количество людей, которые могут описанным образом разделить кокосы. В случае отсутствия решения следует вывести 0.

Примеры

<i>input.txt</i>	<i>output.txt</i>
25	3
20	0

Задача 5. Окружности

Ограничение по времени:

1 секунда на тест

На плоскости задано N различных окружностей. Вам требуется вычислить количество частей, на которые они делят плоскость. В данной задаче будем полагать, что окружность радиуса r с центром в точке x_0, y_0 — это множество точек плоскости $C: C = \{(x, y) \mid (x - x_0)^2 + (y - y_0)^2 = r^2\}$.

Входные данные

Первая строка входного файла содержит число N — количество окружностей ($1 \leq N \leq 50$).

Следующие N строк содержат тройки целых чисел x_0, y_0 и r — координаты центра и радиус каждой из окружностей. Никакие координаты не превышают 10^3 по абсолютной величине, все радиусы положительны и не превышают 10^3 . Никакие две окружности не совпадают.

Выходные данные

В выходной файл необходимо вывести единственное число K — количество частей, на которые окружности делят плоскость.

В связи с ограниченной точностью вычислений с плавающей запятой, не учитываются части плоскости с площадью, меньшей либо равной 10^{-10} .

Пример

<i>input.txt</i>	<i>output.txt</i>
2 0 0 3 0 0 2	3

Задача 6. Платные дороги

Ограничение по времени:

1 секунда на тест

В некоторой стране все дороги платные. Движение по ним установлено только в одну определенную сторону. Между некоторыми городами может быть несколько дорог. Встречаются такие дороги, стоимость проезда по которым отрицательная. Это связано с тем, что расположенные рядом с ними супермаркеты устанавливают дополнительные бонусы своим покупателям. Юный географ Вася получил в школе задание узнать минимальную стоимость проезда между любыми двумя городами этой страны.

Входные данные

В первой строке входного файла заданы числа N и M — количество городов и количество дорог между ними ($1 \leq N \leq 100, 0 \leq M \leq 100000$). Каждая из следующих M строк содержит тройку целых чисел $A B C$, описывающую дорогу из города A в город B , стоимость проезда по которой равна C ($1 \leq A, B \leq N, A \neq B, -1000 \leq C \leq 1000$). Города нумеруются по порядку, начиная с единицы.

Выходные данные

В выходной файл необходимо вывести N строк по N чисел, j -е число в i -й строке — минимальная стоимость проезда из города i в город j . Считаем, что стоимость пути из любого города в него же себя равно 0. Если до какого-то города пути нет, то необходимо вывести символ **z**. Гарантируется, что нет такого циклического пути, стоимость проезда по которому от начала до конца отрицательна.

Пример

<i>input.txt</i>	<i>output.txt</i>
5 7 1 2 1 1 4 -10 4 3 7 4 2 5 2 4 3 2 3 6 5 2 1	0 -5 -3 -10 z z 0 6 3 z z z 0 z z z 5 7 0 z z 1 7 4 0

Задача 7. Ханойские башни

Ограничение по времени:

1 секунда на тест

В задаче о Ханойских башнях требуется на стержень С переместить N дисков, нанизанных на стержень А, пользуясь при этом вспомогательным стержнем В. Все диски имеют разный диаметр и в начальной позиции сложены друг на друга в порядке убывания диаметра снизу вверх. За один ход можно перемещать только один диск — самый верхний на одном из стержней — причем так, чтобы он не оказался нанизанным поверх диска меньшего диаметра.

Задача о Ханойских башнях хорошо изучена. Так, в частности, установлено, что существует единственное решение задачи, состоящее из $2^N - 1$ ходов, и не существует решения, использующего меньшее число перемещений. В классической формулировке задачи $N = 64$, мы же позволим N быть любым натуральным числом, не превышающим 1000.

Для сохранения полного решения задачи при $N = 1000$ не хватило бы всех винчестеров мира. Для печати этого решения на самом быстром принтере потребовалось бы время, в миллиарды раз превышающее возраст Вселенной, а стопка отпечатанных листов бумаги, если бы ее удалось сложить на поверхности Земли, вышла бы далеко за пределы нашей галактики. В связи с этими обстоятельствами у жюри возникли бы определенные трудности с проверкой правильности решений.

Вместо вывода полного решения в выходной файл ваша программа должна проделать следующее. Получив на входе, помимо N , натуральные числа i и j , $i \leq j \leq 2^N - 1$, и считая, что шаги оптимального решения задачи о Ханойских башнях при заданном N пронумерованы натуральными числами $1, 2, \dots, 2^N - 1$, Вы должны вычислить, сколько раз на отрезке от шага i до шага j включительно встречается каждая из букв А, В и С.

Считается, что каждый шаг решения записывается в виде БУКВА \rightarrow БУКВА, например, А \rightarrow В.

Входные данные

В первой строке входного файла записано число N , во второй и третьей — числа i и j , соответственно.

Выходные данные

Выходной файл должен состоять из трех строк, каждая из которых должна содержать по одному целому числу.

В первой строке должно быть записано количество вхождений буквы А, во второй — буквы В, а в третьей — буквы С.

Пример

<i>input.txt</i>	<i>output.txt</i>
7	3
2	2
5	3

Задача 8. Наскальная живопись

Ограничение по времени:

1 секунда на тест

Одним из древнейших наследий человеческой культуры, сохранившимся до наших времен, является наскальная живопись. Первобытные люди запечатлевали на скалах сцены, изображающие события их повседневной жизни — охоту, земледелие, религиозные ритуалы, не подозревая о том, что их рисунки переживут тысячелетия. И в наши дни археологи продолжают открывать все новые и новые изображения, сделанные древними людьми на стенах пещер.

Одна из таких пещер имеет форму прямоугольника размерами $M \times N$ единиц длины. Внутреннее пространство пещеры имеет форму лабиринта. Для простоты будем считать, что вся пещера разделена регулярной сеткой на квадраты размерами 1×1 единиц длины и каждый квадрат либо пуст, либо заполнен монолитным каменным блоком. Квадраты, примыкающие к внешним границам прямоугольника, всегда заполнены такими блоками. На каждой видимой границе каждого блока (то есть, на тех сторонах блоков, которые примыкают к пустому пространству внутри пещеры) находится по одному наскальному рисунку.

В пещеру входит археолог и устанавливает факел точно в центре одной из пустых ячеек. Пусть R, C — номера строки и столбца плана пещеры, на пересечении которых находится эта ячейка (строки и столбцы плана нумеруются, начиная с 0). Свет факела распространяется строго прямолинейно во все стороны. Встречаясь со стеной, свет полностью поглощается, не отражаясь и не проходя насквозь. Таким образом, часть наскальных рисунков будет освещена, а часть останется в темноте. Будем считать, что археолог может рассмотреть наскальный рисунок, если освещено не менее половины той стены, на которой он находится.

Необходимо написать программу, которая по заданным конфигурации лабиринта и местоположению факела находит число наскальных рисунков, которые археолог сможет рассмотреть.

Входные данные

В первой строке входного файла записаны целые числа M, N, R и C ($1 \leq N \leq 100, 0 \leq M \leq 100$). Величина M считается количеством строк плана пещеры, N — количеством столбцов. В последующих M строках располагается план пещеры. Каждая строка плана содержит N символов **X** (заглавная латинская буква «Икс») или **.** (точка). При этом первое означает, что соответствующий квадрат пещеры заполнен каменным блоком, а второе — что он пуст.

Выходные данные

Единственная строка выходного файла должна содержать целое число — количество наскальных рисунков, которые можно рассмотреть.

Пример

<i>input.txt</i>	<i>output.txt</i>
<pre>4 5 2 2 XXXXX X.XXX X...X XXXXX</pre>	8

Задача 9. Как упаковать контейнеры?

Ограничение по времени:

1 секунда на тест

Продукция завода упаковывается в цилиндрические ящики. У всех ящиков одинаковое основание. Высота ящика — неотрицательное целое число, являющееся некоторой степенью числа 2, т.е. оно равно 2^i для некоторого $i = 0, 1, 2, \dots$. Число i (степень двойки) будем называть размером ящика. Все ящики содержат одинаковые товары, но их стоимость может варьироваться. Товары, произведенные раньше, стоят дешевле. Руководство решило, что самые старые (и соответственно, наиболее дешевые) товары должны продаваться в первую очередь. Со склада

товары вывозятся в контейнерах. Контейнеры также цилиндрические, а диаметр каждого контейнера чуть больше диаметра ящика, так что ящики можно легко вставить внутрь контейнеров. Высота контейнера также равна некоторой неотрицательной степени числа 2. Эту степень двойки назовем размером контейнера. Для безопасной транспортировки контейнеры должны быть плотно упакованы, т.е. сумма высот ящиков, помещенных в заданный контейнер, должна быть равна высоте этого контейнера. На склад привезли набор контейнеров. Вам необходимо проверить, можно ли плотно заполнить контейнеры ящиками, имеющимися на складе. Если это возможно, найдите минимальную стоимость товаров, которые можно плотно упаковать в эти контейнеры.

Например, рассмотрим склад с 5 ящиками. Их размеры и стоимость содержимого задана ниже:

Номер ящика	1	2	3	4	5
Размер	1	1	3	2	1
Стоимость	3	2	5	1	4

Два контейнера размера 1 и 2 можно заполнить двумя ящиками суммарной стоимостью 3, 4 или 5, либо тремя суммарной стоимостью равной 9. Контейнер размера 5 нельзя плотно заполнить данными ящиками.

Напишите программу, которая:

- считывает описание ящиков (размер, стоимость) и описание контейнеров (количество имеющихся контейнеров заданного размера) из входного файла;
- проверяет, можно ли плотно заполнить все контейнеры ящиками со склада, и если да, вычисляет минимальную стоимость товаров, которые могут быть упакованы в эти контейнеры;
- записывает результат в выходной файл.

Входные данные

В первой строке входного файла содержится целое число n ($1 \leq n \leq 10000$) – количество ящиков на складе. В каждой из следующих n линий записаны пары неотрицательных чисел, разделенных одним пробелом. Первое из этих чисел задает размер ящика, второе – стоимость товаров в нем. Размер не превышает 1000, а стоимость товаров не превосходит 10000. Следующая строка содержит целое положительное число q , равное количеству различных размеров контейнеров, привезенных на склад. В следующих q строках записано по два положительных числа, разделенных пробелом. Первое число обозначает размер контейнера, а второе – количество контейнеров такого размера. Максимальное число контейнеров равно 5000, а размер контейнеров не превышает 1000.

Выходные данные

Ваша программа должна вывести в первую и единственную строку выходного файла

- единственное слово **NO**, если имеющимися на складе ящиками невозможно плотно заполнить заданный набор контейнеров, либо
- единственное целое число, равное минимальной стоимости товаров в ящиках, которые можно плотно упаковать во все контейнеры.

Пример

<i>input.txt</i>	<i>output.txt</i>
5 1 3 1 2 3 5 2 1 1 4 2 1 1 2 1	3

Задача 10. Строковое уравнение

Ограничение по времени:

1 секунда на тест

Каждая непустая последовательность элементов **0** и **1** называется двоичным словом. Строковое уравнение — это уравнение вида $x_1x_2\dots x_l = y_1y_2\dots y_r$, где x_i и y_j — это двоичные цифры (0 или 1) либо переменные, т.е. маленькие буквы английского алфавита. Для каждой переменной имеется фиксированная длина двоичного слова, которым можно заменить данную переменную. Эта длина называется длиной переменной. Для того чтобы решить уравнение, необходимо назначить каждой переменной двоичное слово соответствующей длины (длина слова, назначенного переменной x должна совпадать с длиной этой переменной) таким образом, что при замене переменных на слова в уравнении, обе части уравнения, которые станут двоичными словами после замены, совпадут.

Для заданного уравнения вычислите, сколько различных решений оно имеет.

Например, пусть a, b, c, d, e — переменные, а 4, 2, 4, 4, 2 — их длины (4 — длина a , 2 — длина b и т.д.). Рассмотрим уравнение: **1bad1 = acbe**. Это уравнение имеет 16 различных решений.

Напишите программу, которая

- читает число уравнений и их описания из входного файла;
- находит количество решений для каждого уравнения;
- выводит результат в выходной файл.

Входные данные

В первой строке входного файла записано число x ($1 \leq x \leq 5$), обозначающее количество уравнений. В следующих строках содержится описание x уравнений. Каждое описание состоит из 6 строк. Между описаниями нет пустых строк. Каждое уравнение описывается следующим образом: в первой строке описания записано число k ($0 \leq k \leq 26$), которое обозначает количество различных переменных в уравнении. Предполагается, что переменные — это первые k маленьких букв английского алфавита. Во второй строке находится последовательность k положительных целых чисел, разделенных одиночными пробелами. Эти числа обозначают длины переменных a, b, \dots в уравнении (первое число — длина a , второе — b , и т.д.). В третьей строке записано число l — длина левой части уравнения, т.е. длина слова, составленного из цифр 0 или 1 и переменных (одиночных букв). Левая часть уравнения записана в следующей строке как последовательность цифр и переменных без пробелов между ними. Следующие две строки содержат описание правой части уравнения. В первой из этих строк записано положительное число r — длина правой части. Вторая строка содержит правую часть уравнения, записанную в том же виде, что и левая часть. Количество цифр плюс сумма длин всех переменных (считаются все вхождения переменных) с каждой стороны уравнения не превышает 10000.

Выходные данные

Для каждого i , $1 \leq i \leq x$, ваша программа должна вывести в выходной файл на отдельной строке количество различных решений i -го уравнения.

Пример

<i>input.txt</i>	<i>output.txt</i>
1 5 4 2 4 4 2 5 1bad1 4 acbe	16