Задача A. Комбинаторная логика

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунд
Ограничение по памяти:	256 мегабайт

Комбинаторная логика — это направление в математической логике, разработанное в первой половине XX века логиками Мозесом Шейнфинкелем и Хаскеллом Карри. Основными элементами комбинаторной логики являются константы, переменные и комбинаторные термы. Термы строятся индуктивно по следующим правилам:

1. если c — константа, то c — комбинаторный терм;
2. если x — переменная, то x — комбинаторный терм;
3. если A и B — комбинаторные термы, то (AB) — комбинаторный терм;
4. других комбинаторных термов нет.

В этом определении выражение (AB) обозначает операцию аппликации — применение функции (комбинатора) A к её аргументу B. Переменные будем обозначать строчными буквами, а комбинаторы заглавными. Константами, в нашем случае, будут только комбинаторы. Для уменьшения количество скобок в записях введено соглашение о том, что пропущенные скобки восстанавливаются по ассоциативности влево:

$$ABC = ((AB)C)$$

Поэтому, будем считать, что «ABC» — это тоже корректная запись комбинаторного терма.

В качестве аксиом возьмём комбинаторы I, K и S, которые определяются по следующим правилам:

- $Ia = a$;
- $Kab = a$;
- $Sabc = ac(bc)$.

Мы не будем сильно углубляться в теорию, поэтому для вычисления значения комбинаторов будем действовать по следующим правилам:

- Для вычисления комбинатора I возьмём один терм, следующий за ним, для K — 2 терма, для S — 3. Если термов недостаточно, то строка не будет больше изменяться.
- Каждый раз вычисляем либо самый левый комбинатор, либо самый левый комбинатор в какой–нибудь скобке.
- Будем опускать левые скобки, когда они есть.

Примеры:

$$S(KII)ab = S(I)ab = (I)b(ab) = b(ab);$$

$$SIISa = IS(IS)a = SSa;$$

$$S(KK)Iab = (KK)a(Ia)b = KKa(Ia)b = K(Ia)b = Kab = a.$$

Дан комбинатор P, записанный через комбинаторы I, K, S. Найдите значение после применения его к последовательности, составленной из первых n строчных букв латинского алфавита.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leqslant n \leqslant 8$).

Вторая строка содержит описания комбинатора P — строку, состоящую из символов «I», «K», «S», «(» и «)». Гарантируется, что P — комбинаторный терм. Длина строки не меньше 1 и не больше 150 000. Максимальная вложенность скобок не превосходит 100. Общее количество комбинаторов в процессе вычисления не превзойдёт 50 000.

Формат выходного файла

Выходной файл должен содержать одну строку — результат применения комбинатора P к последовательности, составленной из первых n латинских букв. Гарантируется, что длина строки не превзойдёт 8, и она будет состоять только из строчных латинских букв.

Примеры

input.txt	output.txt
1 I	a
2 K	a
1 SII	aa
1 S(SII)I	aaa
3 S((S(KS)K)(S(KS)K)S)(KK)	acb
2 SS(K(SKK))	abb

Задача B. Последовательности

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунд
Ограничение по памяти:	256 мегабайт

Рассмотрим последовательность из n символов. Скажем, что она k-простая, если на позициях i и $i+k$ стоят разные символы, при $i = 1, 2, ..., n-k$. Найдите количество k-простых последовательностей, состоящих из первых m латинских букв.

Формат входного файла

Входной файл содержит три целых числа n ($1 \leqslant n \leqslant 40$), m ($1 \leqslant m \leqslant 8$) и k ($1 \leqslant k \leqslant 8$).

Формат выходного файла

Выходной файл должен содержать одно целое число — количество таких последовательностей.

Примеры

input.txt	output.txt
3 2 2	4

В примере этими последовательностями будут: `aab`, `abb`, `baa`, `bba`.

Задача C. Игра в камни

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

Два игрока играют в следующую игру. Имеется n кучек камней, в i-й кучке a_i камней. Игроки делают ходы по очереди. На каждом ходу игрок выбирает кучку, и убирает из неё любое положительное число камней. Проигрывает тот, кто возьмёт последний камень. Для данной позиции определите, выигрышная ли она для игрока, который начинает игру, и если да, то какой ход ему следует сделать, чтобы выиграть независимо от ходов соперника.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leqslant n \leqslant 100$) — количество кучек. Вторая строка содержит n целых чисел a_i ($1 \leqslant a_i \leqslant 100$) — количество камней в i-й кучке.

Формат выходного файла

Первая строка выходного файла должна содержать строку «`Lose`» — если при наилучшей стратегии обоих игроков, игрок, начинающий игру проиграет, или «`Win`», если позиция выигрышная. В случае выигрышной позиции, вторая строка должна содержать два целых числа — номер кучки и количество камней, которые надо из неё взять соответственно.

Примеры

input.txt	output.txt
2 3 3	Lose
2 3 2	Win 1 1
3 3 3 3	Win 1 3

Задача D. Одинаковые вершины

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунда
Ограничение по памяти:	256 мегабайт

Сравните все вершины ориентированного графа, в котором каждая вершина имеет метку и не более одного исходящего ребра.

Две вершины называются одинаковыми, если они неотличимы следующим алгоритмом:

1. Сравнить метки. Если они не равны, алгоритм завершается и сообщает, что вершины различны.
2. Сравнить исходящие степени. Если они не равны, алгоритм завершается и сообщает, что вершины различны.
3. Если обе вершины не имеют исходящих ребер, алгоритм завершается и сообщает, что вершины одинаковые.
4. Повторить алгоритм для вершин, на которые указывают исходящие ребра.

Обратите внимание на третий пункт: если алгоритм останавливается на нем и сообщает, что вершины одинаковые, то это подразумевает, что вершины неотличимы.

Формат входного файла

В первой строке входного файла записано единственное число n ($2 \leqslant n \leqslant 10^5$) — количество вершин.

В следующих n строках описаны вершины графа. Каждое описание состоит из строчной латинской буквы (метка вершины) и числа t, разделенных пробелом. Если $t = 0$, то вершина не имеет исходящих ребер. В противном случае ребро ведет в t-ую вершину. Вершины пронумерованы от 1 до n в том же порядке, в каком они следуют во входном файле.

Формат выходного файла

Выходной файл должен содержать ровно n строк.

В i-ой строке должен быть записан номер класса, которому принадлежит i-ая вершина. Одинаковые вершины должны принадлежать одному классу, различные — разным.

Первая вершина должна принадлежать первому классу. Следующая вершина, не принадлежащая первому классу, должна принадлежать второму, и так далее.

Пример

input.txt	output.txt
4 a 2 b 3 b 0 b 3	1 2 3 2

Задача E. Пинбол

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	4 секунд
Ограничение по памяти:	256 мегабайт

Пинбол — аркадная игра, в которой игрок набирает игровые очки, манипулируя одним металлическим шариком на игровом поле при помощи флипперов. Основная цель игры — набрать максимальное количество очков.

Поле содержит множество различный объектов, но в нашей задаче ограничемся только бамперами. Геометрически, бампер — круг определённого радиуса. При соприкосновении металлического шарика с бампером игрок получает некоторое количество очков, а сам шарик отлетает от него в любую сторону (возможно, даже, в ту, в направлении которой он двигался) и направляется к другому бамперу. Шарик может отскочить только к такому бамперу, что, если провести отрезок между их центрами, то он не пересечёт и не будет касаться других бамперов. Причём, шарик не может отскочить к тому же бамперу, от которого он прилетел (формально говоря, если шарик сначала столкнулся с бампером a, потом с бампером b, то он не сможет после последнего соприкосновения снова направится к a).

Пусть центр 1-го бампера имеет координаты (x_1, y_1), и радиус 1-го бампера равен r_1, центр и радиус второго 2-го — (x_2, y_2), r_2 соответственно. Тогда время, которое нужно затратить шарику, чтобы преодолеть расстояние от первого бампера до второго будет равно $\lceil \frac{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}-r_1-r_2}{20} \rceil$ секундам (дробь нужно округлить вверх до ближайшего целого). Будем считать, что в момент времени 0 игра только началась, и шарик столкнулся с самым верхним бампером (с бампером с максимальной координатой y центра бампера, если таких оказалось несколько, тогда он столкнулся с бампером имеющим эту же координату y и наибольшую координату x).

Напишите программу, которая по данной конструкции игрового поля, будет обрабатывать 2 вида запросов.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($3 \leqslant n \leqslant 7$) — количество бамперов. Далее следуют n строк, содержащие по 4 целых числа x_i, y_i, r_i и s_i ($0 \leqslant x_i, y_i \leqslant 50$, $1 \leqslant r_i \leqslant 5$, $-100 \leqslant s_i \leqslant 100$), означающий, что i-й бампер имеет координаты (x_i, y_i), радиус r_i, а при соприкосновении шарика с ним игрок получит s_i очков. Бамперы не пересекаются и не касаются друг друга. Гарантируется, что не возникнет ситуации, когда шарику будет некуда лететь.

Следующая строка содержит одно целое число q ($1 \leqslant q \leqslant 10\,000$) — количество запросов. Далее следуют q строк. Первое целое в строке число означает тип запроса. Если это первый тип, то далее следует одно целое число t_j ($0 \leqslant t_j \leqslant 10^9$). Если это второй тип, то далее следует два целых числа t_j и x_j ($0 \leqslant t_j \leqslant 10^9$, $1 \leqslant x_j \leqslant n$).

Формат выходного файла

Для каждого запроса первого типа, выведите одно целое число — максимальное количество очков, которое можно набрать в процессе игры, если играть первые t_i секунд.

Для каждого запроса второго типа, выведите одно целое число — максимальное количество очков, которое можно набрать на t_j секунде игры, если последним был задет x_j-й бампер. Если такая ситуация не достижима, то выведите один символ — «#».

Все ответы на запросы должны быть в отдельных строках.

Примеры

input.txt	output.txt
3 0 0 1 1 50 0 1 1 0 50 1 1 20 1 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 2 1 1 2 2 1 2 3 1 2 4 1 2 5 1 2 6 1 2 7 1 2 8 1 2 9 1 2 10 1	1 1 2 2 2 3 3 3 3 4 # # 2 2 2 # 3 3 3 #

Задача F. Комбинаторная логика 2

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунд
Ограничение по памяти:	256 мегабайт

Комбинаторная логика — это направление в математической логике, разработанное в первой половине XX века логиками Мозесом Шейнфинкелем и Хаскеллом Карри. Основными элементами комбинаторной логики являются константы, переменные и комбинаторные термы. Термы строятся индуктивно по следующим правилам:

1. если c — константа, то c — комбинаторный терм;
2. если x — переменная, то x — комбинаторный терм;
3. если A и B — комбинаторные термы, то (AB) — комбинаторный терм;
4. других комбинаторных термов нет.

В этом определении выражение (AB) обозначает операцию аппликации — применение функции (комбинатора) A к её аргументу B. Переменные будем обозначать строчными буквами, а комбинаторы заглавными. Константами, в нашем случае, будут только комбинаторы. Для уменьшения количество скобок в записях введено соглашение о том, что пропущенные скобки восстанавливаются по ассоциативности влево:

$$ABC = ((AB)C)$$

Поэтому, будем считать, что «ABC» — это тоже корректная запись комбинаторного терма.

В качестве аксиом возьмём комбинаторы I, K и S, которые определяются по следующим правилам:

- $Ia = a$;
- $Kab = a$;
- $Sabc = ac(bc)$.

Мы не будем сильно углубляться в теорию, поэтому для вычисления значения комбинаторов будем действовать по следующим правилам:

- Для вычисления комбинатора I возьмём один терм, следующий за ним, для K — 2 терма, для S — 3. Если термов недостаточно, то строка не будет больше изменяться.
- Каждый раз вычисляем либо самый левый комбинатор, либо самый левый комбинатор в какой–нибудь скобке.
- Будем опускать левые скобки, когда они есть.

Примеры:

$$S(KII)ab = S(I)ab = (I)b(ab) = b(ab);$$

$$SIISa = IS(IS)a = SSa;$$

$$S(KK)Iab = (KK)a(Ia)b = KKa(Ia)b = K(Ia)b = Kab = a.$$

Пусть мы применили комбинатор P к последовательности, состоящей из первых n строчных латинских букв, и получили последовательность X. По последовательности X и числу n выразите каким–нибудь образом комбинатор P через комбинаторы I, K и S.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leqslant n \leqslant 8$). Вторая строка содержит непустую последовательность до 8 символов. Она может содержать только первые n строчных букв латинского алфавита.

Формат выходного файла

Выходной файл должен содержать одну строку, содержащую символы «I», «K», «S», «(» и «)» — описание комбинаторного терма P. Длина выводимой строки не должна превышать 400 000. Максимальная вложенность скобок не должна превышать 150. Общее количество комбинаторов в процессе вычисления не должно превзойти 150 000. Гарантируется, что ответ всегда существует.

Примеры

input.txt	output.txt
1 a	I
2 a	K
1 aa	SII
1 aaa	S(SII)I
3 acb	S((S(KS)K)(S(KS)K)S)(KK)
2 abb	SS(K(SKK))

Задача G. Тетраэдр минимального объема

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

Дан трехгранный угол $OABC$ с вершиной в точке $O(0,0,0)$ и с точками $A(a_1,a_2,a_3)$, $B(b_1,b_2,b_3)$, $C(c_1,c_2,c_3)$ на его ребрах (a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3 -- целые положительные числа). Координаты точек указаны в декартовой системе координат. Точка $P(p_1,p_2,p_3)$, где p_1, p_2, p_3 — целые положительные числа, лежит строго внутри трехгранного угла (т.е. вектор OP имеет в базисе OA, OB, OC положительные координаты). Провести через точку P плоскость так, чтобы она отсекала от данного трехгранного угла тетраэдр наименьшего объема.

Формат входного файла

В четырех строках входного файле через пробел записаны координаты точек A, B, C, P соответственно. Координаты — целые положительные числа, не превосходящие 10^3. Вектора OA, OB и OC линейно независимы.

Формат выходного файла

Выходной файл должен содержать одно число — наименьший объем тетраэдра, отсекаемого плоскостью от данного трехгранного угла, с точностью до 10^{-5}.

Примеры

input.txt	output.txt
1 2 3 2 3 1 2 5 3 2 4 3	2.53125

Задача H. Вписанный треугольник 3

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

Возьмём произвольный невырожденный треугольник. Проведём замкнутую ломанную из трёх звеньев минимальной длины так, чтобы на каждой стороне треугольника лежала хотя бы одна вершина ломанной, и каждая вершина ломаной лежала на какой–нибудь стороне. Определите, может ли эта ломанная иметь длины звеньев равными a, b и c?

Формат входного файла

Входной файл содержит три целых числа a, b и c ($-1\,000 \leqslant a, b, c \leqslant 1\,000$).

Формат выходного файла

Выходной файл должен содержать одну строку «`Yes`», если звенья ломанной могут иметь заданные длины, и «`No`» в противном случае.

Примеры

input.txt	output.txt
2 3 4	Yes
-1 -1 -1	No

Задача I. $A^2 + \ldots + B^2$

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунда
Ограничение по памяти:	256 мегабайт

$$N = \sum_{k=A}^{B} k^2$$

Формат входного файла

В единственной строке записаны два целых числа A и B ($-10^{10^6} \leqslant A \leqslant B \leqslant 10^{10^6}$), разделенных пробелом.

Формат выходного файла

Выведите в единственной строке цифровой корень[1] числа N.

Пример

input.txt	output.txt
1 2	5

[1] Цифровой корень числа равен самому числу, если число состоит из одной цифры, и равен цифровому корню суммы цифр этого числа в противном случае.

Задача J. Дни рождения

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Миша хочет устроиться на работу в отдел, где работает Коля. Он ничего не знает о количестве сотрудников фирмы и считает, что число сотрудников фирмы может быть любым с равной веронятностью.

Чтобы проверить математические способности Миши, Коля предложил ему определить количетство сотрудников фирмы, предоставив Мише дополнительную информацию о количестве совпадений дней рождений сотрудников фирмы.

Всех сотрудников можно разбить на группы, так что внутри группы у всех сотрудников дни рождения совпадают, а у сотрудников из разных групп дни рождения не совпадают. Оказалось, что среди этих групп ровно m_2 групп состоит из двух человек, ровно m_3 групп состоит из трех человек, ... , ровно m_s групп состоит из s человек, а все остальные группы состоят из одного человека.

Считая, что в любом году d дней, и что вероятность родиться в любой из дней года одинакова, помогите найти Мише наиболее вероятное число сотрудников фирмы.

Формат входного файла

Первая строка входного файла содержит два целых числа s и d ($365 \leqslant d \leqslant 40\,000$, $2 \leqslant s \leqslant d$). Вторая строка содержит $s - 1$ целых чисел m_2, m_3, ..., m_s ($0 \leqslant \sum_{i=2}^{s} m_i \leqslant d$).

Формат выходного файла

Выходной файл должен содержать одно положительное целое число n — наиболее вероятное число сотрудников. Если таких n несколько, то выаедите ответ с максимальным n.

Примеры

input.txt	output.txt
2 365 1	28
4 365 3 5 2	113

Задача K. Радиовышки

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	8 секунд
Ограничение по памяти:	256 мегабайта

На одной из военных баз столкнулись с непредвиденной проблемой. В поле были размещены здания, но не было никакой связи между ними. Для устранения этого, было решено поставить на этом поле несколько радиовышек и по одной радиовышке на каждое здание. Тогда сигналы можно будет передавать от одной радиовышки до другой. Но, так как это дело не первой необходимости, то денег на вышки надо потратить как можно меньше.

Более формально. Дана таблица размером $n \times m$. Каждая ячейка либо пуста, либо содержит здание. В любую пустую ячейку можно поставить радиовышку, и, обязательно, на каждое здание требуется установить по радиовышке. Каждая радиовышка имеет мощность от 1 до 9. Это число означает, на каком расстоянии она будет работать (отправлять сигналы до других радиовышек). Расстояние считать Евклидовым. Стоимость установки одной вышки равна a. Если вышка имеет мощность p, то надо дополнительно заплатить p^2 единиц.

Определите, где и какие вышки необходимо разместить, чтобы можно было передать сигнал от любого здания к любому другому (возможно не напрямую), и чтобы стоимость размещения вышек была минимальна.

Формат входного файла

Первая строка входного файла содержит 2 целых числа n и m ($1 \leqslant n, m \leqslant 20$, $2 \leqslant n \times m \leqslant 25$) — размер поля. Далее следуют n строк по m симолов — описание поля: «`.`» — пустая ячейка, «`*`» — ячейка, занятая зданием. Гарантируется, что на поле всегда будут пресутсвовать по крайней мере 2 здания.

Последняя строка входного файла содержит одно целое число a ($1 \leqslant a \leqslant 500$).

Формат выходного файла

Выходной файл должен содержать n строк по m символов — поле, после установки вышек: «`.`» — пустая ячейка, «`x`» — ячейка, занятая вышкой мощностью x, где x — целое число от 1 до 9.

Примеры

input.txt	output.txt
3 3 ..* ... *.. 4	..2 .2. 2..
5 4 *... ...* .... *... ...* 9	1... 3..3 .... 2... ...3