

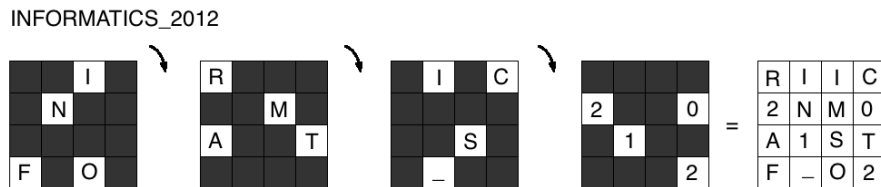
Задача А. Кардано

Имя входного файла: A.in
Имя выходного файла: A.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

Решётка Кардано — инструмент шифрования, представляющий собой специальную квадратную таблицу-карточку размера $N \times N$, часть ячеек которой вырезана. Длина сообщения, которое нужно зашифровать, должна быть равна N^2 символов. Решетка Кардано накладывается на бумагу и сообщение выписывается по одному символу в вырезанную ячейку. Когда вырезанные ячейки окажутся заполнены решетка поворачивается на 90 градусов по часовой стрелке и процесс продолжается. Так повторяется еще 2 раза.

Назовем решетку “правильной”, если при шифровании ни один символ не наложится на другой и “неправильной” в противном случае. Назовем “правильную” решетку “хорошей”, если все N^2 символов исходного сообщения будут записаны и “плохой” в противном случае.

Для заданных решеток, определите, правильные, ли они, а для правильных — хорошие или плохие.



Формат входного файла

Первая строка входного файла содержит одно целое число K ($1 \leq K \leq 10$) — количество решеток для проверки. Затем следует описание K решеток. Описание каждой решетки начинается со строки, содержащей целое число N ($1 \leq N \leq 100$) — размер решетки. Затем следует N строк, содержащих по N символов каждая. Символ “*” означает невырезанную часть решетки, “.” — вырезанную.

Формат выходного файла

Выходной файл должен содержать K строк — по одной для каждой таблицы. Если соответствующая решетка неправильная, выведите “INCORRECT”, если плохая — “BAD”, если хорошая — “GOOD”.

Примеры

A.in	A.out
3	INCORRECT
2	BAD
.*	GOOD
*.	
4	
.***	

**.*	

4	
**.*	
*.**	

.**.	

Задача В. Строчная арифметика

Имя входного файла: B.in
Имя выходного файла: B.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

Для двух строк $S = s_1s_2s_3\dots s_n$ и $T = t_1t_2t_3\dots t_m$ определим операцию сложения следующим образом: $R = S + T = s_1s_2\dots s_nt_1t_2\dots t_m$, а операцию умножения следующим образом: $R = S \cdot T = s_1t_1s_1t_2s_1t_3\dots s_1t_ms_2t_1\dots s_2t_m\dots s_nt_1\dots s_nt_m$. Заметьте, что в общем случае $(S \cdot T) \cdot R \neq S \cdot (T \cdot R)$, поэтому операции умножения нужно выполнять последовательно слева направо. Как обычно, операция умножения имеет больший приоритет.

Дается выражение на строках с операциями сложения и умножения. Посчитайте количество символов в строке-результате.

Формат входного файла

Единственная строка входного файла содержит арифметическое выражение, использующее операции сложения и умножения, а в качестве аргументов — строки, состоящие из строчных букв английского алфавита. Строка не содержит пробелов и ее длина не превышает 100 символов. В строке присутствуют только символы “+”, “*” и строчные буквы английского алфавита.

Формат выходного файла

Выведите одно число без ведущих нулей — ответ к задаче.

Примеры

B.in	B.out
ab*cd*ef+g*h	34

Note

В 70% тестов ответ не превышает 10^{18}

Задача С. Быки и коровы

Имя входного файла: --
Имя выходного файла: --
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

В этой задаче Вам предлагается сыграть в известную игру “Быки и коровы”. Компьютер загадывает число из 4-х различных цифр от 1 до 9, а Ваша программа должна его отгадать с нескольких попыток. Попытка — это 4-значное число с неповторяющимися цифрами. В ответ на попытку Вашей программе будет даваться ответ в виде целого числа, равного $10 \cdot X + Y$, где X — количество цифр, угаданных на неправильных позициях, а Y — количество цифр, угаданных на верных позициях. Таким образом, в случае если программа правильно угадала загаданное число, код ответа будет равен 4. В этом случае программа должна завершиться. Учтите, что количество потраченных попыток будет учитываться при подсчете оценки за тест.

Ваша программа должна делать попытки, вызывая функцию `guess(x)`, параметром и результатом которой является целое число. Повторный вызов `guess` с аргументом, который ранее использовался, либо передача в качестве аргумента не 4-х значного числа, либо числа, содержащего одинаковые цифры, либо числа, содержащего цифру 0, считается ошибкой.

Для правильной работы Вашей программы необходимо подключить библиотеку/модуль `bullscows`:

C/C++: `#include "bullscows.h"`

Pascal: `uses bullscows;`

Функция объявлена следующим образом:

C/C++: `int guess(int x);` Pascal: `function guess(x : longint) : longint;`

Оценка будет происходить следующим образом. Пусть T — полный балл за тест, а R — балл, полученный программой. Если Ваша программа угадывает число не более чем за 7 попыток, то $R = T$, если больше чем за 7, но не больше, чем за 10, то $R = 0.75 \cdot T$, если больше чем за 10, но не больше, чем за 14, то $R = 0.5 \cdot T$, если больше чем за 14, но не больше, чем за 30, то $R = 0.25 \cdot T$. В остальных случаях $R = 0$.

Задача D. Клуб читателей

Имя входного файла: D.in
Имя выходного файла: D.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

В клубе состоят N человек. Каждый месяц выходит новая книга, и Вы рассылаете ее членам клуба. Так как покупать всем по книге выйдет очень дорого, Вы можете разослать книги только определенным читателям. Читатели, получившие книгу снимают с нее несколько копий и рассылают всем своим хорошим знакомым в клубе. Читатели, получившие копию книги, делают то же самое. Если читатель A является хорошим знакомым читателя B , это не означает, что B является хорошим знакомым A .

Вам известны все хорошие знакомые каждого читателя. Определите минимальное количество экземпляров книги, которое необходимо приобрести и отправить, чтобы все читатели клуба получили бы книгу или ее копию.

Формат входного файла

В первой строке входного файла задаются два целых числа N и M ($1 \leq N \leq 10000$, $1 \leq M \leq 100000$). В следующих M строках задаются по два целых числа A и B : читатель с номером B — хороший знакомый читателя с номером A ($1 \leq A, B \leq N$, $A \neq B$). Числа в строках разделены пробелами.

Формат выходного файла

В первой строке выходного файла выведите минимальное количество экземпляров книги, которое необходимо отправить. Во второй строке — номера читателей, которые должны их получить. Если вариантов несколько, выведите любой.

Примеры

D.in	D.out
5 5 1 2 1 3 3 1 2 4 5 4	2 5 3
5 4 1 2 3 2 2 4 2 5	2 3 1

Задача Е. Медкомиссия

Имя входного файла: E.in
Имя выходного файла: E.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

Начался весенний призыв и военкоматы заполнились призывниками. Начальникам военкоматов поступил запрос о продолжительности проведения медкомиссий. Один из них обратился к Вам за помощью.

Каждый из N призывников должен пройти M медицинских кабинетов. В одном кабинете одновременно могут осматривать только одного призывника. После завершения осмотра в i -м кабинете призывник мгновенно попадает в очередь в $i + 1$ -й кабинет. Как только в каком-то кабинете освобождается место, его мгновенно заполняет призывник, стоящий первым в очереди в этот кабинет. После завершения осмотра в M -м кабинете призывник отправляется домой собирать вещи. Продолжительностью медкомиссии считается время между входом первого призывника в первый кабинет и выходом последнего призывника из M -го кабинета.

Зная для каждого призывника время, которое он будет находиться в каждом кабинете (вы заранее обработали информацию из медицинской карточки), а также порядок входа в первый кабинет, определите продолжительность медкомиссии.

Формат входного файла

Первая строка входного файла содержит 2 целых числа N, M ($1 \leq N \leq 10^5, 1 \leq M \leq 10$). Каждая из следующих N строк содержит по M целых чисел. j -е число на i -й строке — время обследования i -го призывника в j -м кабинете ($1 \leq i \leq N, 1 \leq j \leq M$). Время — целое число в интервале от 1 до 1000. Призывники нумеруются в порядке входа в первый кабинет.

Формат выходного файла

Выведите одно целое число — продолжительность проведения медкомиссии.

Примеры

E.in	E.out
3 2 3 5 4 4 5 3	15
3 1 3 4 5	12

Задача F. ДНК

Имя входного файла: `F.in`
Имя выходного файла: `F.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

Одна из задач генетического анализа заключается в определении степени похожести двух последовательностей нуклеотидов. Последовательность нуклеотидов - это строка из букв A, G, C, T. Последовательности можно циклически сдвигать друг относительно друга. Степенью похожести двух последовательностей назовем максимально возможное количество совпадений символов в соответствующих позициях строк. Для заданных двух последовательностей определите степень их похожести.

Формат входного файла

Входной файл содержит две строки одинаковой длины, состоящие из символов A, G, C, T. Строки не пустые и их длина не превышает 50000.

Формат выходного файла

На первой строке выходного файла выведите целое число — степень похожести заданных строк. На следующих двух строках выведите исходные строки, циклически сдвинутые так, что достигается полученная степень похожести. Строки выводите в таком же порядке, как они даны во входном файле.

Примеры

F.in	F.out
ACAGTG	5
AGTGTC	ACAGTG
	TCAGTG

Note

В не менее 50% тестов длина каждой строки не превышает 10000

Задача G. Простая задача

Имя входного файла: `G.in`
Имя выходного файла: `G.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

Имеется массив из N целых чисел. Могут поступать следующие запросы:

- `S a b` — записать в ячейку с номером a значение b ($1 \leq a \leq N$, $|b| \leq 1000000$).
- `R l r` — перевернуть подмассив с позиции l до позиции r ($1 \leq l \leq r \leq N$).
- `Q l r` — вывести сумму элементов с позиции l до позиции r ($1 \leq l \leq r \leq N$).

Формат входного файла

В первой строке входного файла дано одно целое число N ($1 \leq N \leq 100000$). В следующей строке даны N чисел, каждое из которых по абсолютному значению не превышает 1000000. В следующей строке дается одно целое число M — количество запросов ($1 \leq M \leq 100000$). В следующих M строках заданы запросы в том виде, в каком они описаны в условии.

Формат выходного файла

Для каждого запроса, начинающегося с `Q`, выведите одну строку — ответ на этот запрос.

Примеры

G.in	G.out
4	6
1 2 3 4	11
4	
Q 1 3	
R 2 4	
S 1 4	
Q 1 3	

Note

В не менее 40% тестов $N, M \leq 1000$

Задача Н. Казарма

Имя входного файла: `H.in`
Имя выходного файла: `H.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

В казарме N рядовых солдат. Когда в казарму входит старший сержант, они строятся в ряд. После построения старший сержант идет от первого солдата к последнему и по своему желанию выбирает троих. Эти солдаты делают шаг вперед. Если они стоят в порядке возрастания роста, то у командира хорошее настроение, и он отпускает всех. Если он никак не может выбрать трех солдат так, чтобы у него было хорошее настроение, он злится и заставляет всех бегать весь день.

Известно, что у всех солдат рост разный. Один любопытный солдат хочет знать, сколькими способами все солдаты могут встать в строй так, чтобы старший сержант был доволен. Так как ответ может быть большим, выведите его по модулю M .

Формат входного файла

В единственной строке входного файла задаются два целых числа N и M , разделенных пробелом ($3 \leq N \leq 25000$, $1 \leq M \leq 2 \cdot 10^9$).

Формат выходного файла

Выведите ответ к задаче.

Примеры

<code>H.in</code>	<code>H.out</code>
3 10000	1
5 10000	78

Note

- В не менее 20% тестов $N \leq 10$
- В не менее 50% тестов $N \leq 500$
- В не менее 75% тестов $N \leq 4000$