

Задача А. Анализ растительности

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Деревом называется связный граф без циклов (в частности, изолированная вершина тоже является деревом). **Лесом** называется граф, состоящий из нескольких непересекающихся деревьев.

Дан лес с N вершинами и M рёбрами. Из какого наибольшего количества деревьев он может состоять?

Формат входных данных

Первая строка входных данных содержит два целых числа N и M ($2 \leq M, N \leq 10^4$).

Формат выходных данных

Выведите одно целое число — наибольшее количество деревьев, из которого может состоять лес. Если при данных M и N граф не может быть лесом, выведите 0.

Примеры

стандартный ввод	стандартный вывод
4 2	2
1201 2020	0

Задача В. Битовые последовательности

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Последовательность будем называть разнообразной, если в ней нет ни одной пары равных соседних элементов.

Для заданного неотрицательного N определить длину самой длинной разнообразной подстроки двоичной записи этого числа с не менее, чем одним ведущим нулём.

Заметим, что, согласно приведенному определению, любую подстроку длины 1 можно считать разнообразной.

Формат входных данных

Входной файл содержит одно целое число N ($1 \leq N \leq 10^{19}$).

Формат выходных данных

Выведите одно целое число — ответ к задаче.

Примеры

стандартный ввод	стандартный вывод
25	2
2	3
26	4

Задача С. Выгибание многоугольников

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Кати есть проволока целочисленной длины N . Она хочет согнуть её в нескольких местах так, чтобы получился правильный многоугольник с целой длиной стороны. Сколькими способами Катя сможет это сделать?

Формат входных данных

Входные данные содержат одно целое число N ($1 \leq N \leq 10^9$).

Формат выходных данных

Выведите одно целое число — количество различных правильных многоугольников, которые сможет получить Катя.

Примеры

стандартный ввод	стандартный вывод
2	8
3	9

Задача D. Генерация паролей

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В современных системах при генерации паролей часто пытаются добиться запоминаемости.

В новой операционной системе «Унас» все пароли являются кириллическими. Используются 33 буквы кириллицы, из них 10 букв являются гласными, 21 — согласными; твёрдый и мягкий знаки являются специальными буквами.

Для запоминаемости используются следующие правила:

- Мягкий и твёрдый знак можно использовать только после согласных (кроме «й»); эти знаки не могут идти перед согласной.
- Рядом с гласными не должно быть других гласных, рядом с согласными — других согласных.

Вычислите, какое количество паролей длины N является запоминающимися.

Формат входных данных

Входные данные содержат одно целое число N ($6 \leq N \leq 10^6$) — длина пароля.

Формат выходных данных

Выведите одно целое число — остаток от деления количества запоминающихся паролей длины N на $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
2	460

Задача Е. Длительность выполнения

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В языке программирования Bracket программа является регулярным скобочным выражением.

Формально программа определяется следующим образом. Если строка является корректной программой на языке Bracket, выведите время её выполнения. В противном случае выведите строку «`Compilation error`» и номер первой позиции с ошибкой. (здесь и далее угловые скобки являются частью описания синтаксиса Bracket, но не являются частью самого языка):

$\langle \text{prog} \rangle ::= ' ' | (\langle \text{prog} \rangle) | [\langle \text{prog} \rangle] | \{\langle \text{prog} \rangle\} | \langle \text{prog} \rangle \langle \text{prog} \rangle$

Здесь ' ' обозначает пустую строку.

Время на выполнение программ на языке Bracket вычисляется следующим образом: пусть A — любая программа на Bracket, B и C — непустые программы на Bracket, тогда время выполнения вычисляется по следующим правилам:

1. $T \langle ' ' \rangle = 1$
2. $T \langle (A) \rangle = T \langle A \rangle + 1$
3. $T \langle [A] \rangle = T \langle A \rangle + 2$
4. $T \langle \{A\} \rangle = T \langle A \rangle + 3$
5. $T \langle BC \rangle = T \langle B \rangle \cdot T \langle C \rangle$.

Для заданной строки проверить, является ли она корректной программой на языке Bracket, и, если является, вычислить время выполнения этой программы; в противном случае выдать текст «`Compilation error`», за которым через пробел следует номер первой позиции, в которой гарантированно есть ошибка (позиции в Bracket нумеруются слева направо, начиная с 1). Очевидно, что если программа закончилась без ошибок, но осталась хотя бы одна незакрытая скобка, ошибкой считается позиция с номером, равным длине строке плюс 1.

Формат входных данных

Строка длины не более 60 символов, составленная из круглых, квадратных и фигурных скобок.

Формат выходных данных

Если строка является корректной программой на языке Bracket, выведите время её выполнения. В противном случае выведите строку «`Compilation error`» и номер первой позиции с ошибкой.

Примеры

стандартный ввод	стандартный вывод
	1
()	2
[]	3
() [(1)]	Compilation error 5
([] () [([] ())])	Compilation error 13
([[] ()] () [])	49
{}	4
([] {})	13

Задача F. Есть четыре команды...

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В операционной системе «RebootOS» процессы могут быть либо активными, либо выгруженными на диск. Пользователь всегда работает с активным процессом с наименьшим номером. Всего в системе N процессов, пронумерованных последовательными целыми числами от 1 до N .

У пользователя есть несколько команд:

1. Активировать процесс с номером 1. Если процесс с номером 1 активен, ничего не происходит.
2. Выгрузить процесс с номером 1. Если процесс с номером 1 выгружен, ничего не происходит.
3. Из активного процесса с наименьшим номером выгрузить следующий процесс в своп. Если этот процесс уже выгружен или следующего процесса нет, ничего не происходит.
4. Из активного процесса с наименьшим номером сделать активным следующий процесс. Если этот процесс уже активен или следующего процесса нет, ничего не происходит.

Известно текущее состояние процессов. Требуется за наименьшее количество команд перевести систему в заданное состояние.

Формат входных данных

Первая строка входных данных содержит одно целое число N ($1 \leq N \leq 50$). Вторая строка задаёт текущее состояние процессов и состоит из N символов 0 и 1. 0 в i -й слева позиции обозначает, что i -й процесс выгружен, 1 — что процесс активен. Третья строка задаёт требуемое состояние процессов в том же формате.

Формат выходных данных

Выведите одно целое число — минимальное количество команд, требуемое для того, чтобы перевести текущее состояние процессов в заданное.

Пример

стандартный ввод	стандартный вывод
3 111 000	5

Замечание

Порядок операций: 4 (011), 4 (010), 1 (110), 4 (100), 2 (000).

Задача G. Ё-компьютер

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В разработанном в СССР в 50-х годах прошлого века компьютере «Сетунь» использовалась сбалансированная троичная система счисления.

Сейчас в рамках импортозамещения разрабатывается «Ё-компьютер», основанный на похожих идеях. Учитывая изменения в качестве подготовки программистов с советских времён, компьютер должен будет работать уже с обычными троичными числами.

Среди директив процессора — групповое поразрядное сложение. Для двух заданных неотрицательных троичных чисел требуется определить троичное число, полученное поразрядным сложением по модулю 3 всех чисел, расположенных между данными числами, включая и эти числа.

Формат входных данных

Входной файл состоит из двух непустых строк. В первой строке задаётся первое число, во второй строке — второе число. Строки состоят из цифр 0,1,2 и не содержат лишних ведущих нулей. Длина каждой строки не превышает 250 000.

Формат выходных данных

Выведите получившуюся строку без ведущих нулей.

Примеры

стандартный ввод	стандартный вывод
12 120	102
112 12	12

Задача Н. Жёсткий фактчекинг

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

На одной из страниц в популярной социальной сети HackrHackr пользователь Вася Пупкин разместил несколько селфи на фоне небоскрёбов, подписав его «Я в Эпл-Сити».

Посетители страницы усомнились в том, что на фотографии действительно изображён Эпл-Сити, и решили провести проверку. Для этого они измерили высоту всех небоскрёбов на фотографии (небоскрёбы можно представить прямоугольниками со стороной длины 1 и высоты h_i , стоящими вплотную друг к другу; таким образом, каждая фотография задаётся как последовательность высот h_i), после чего нашли описание улицы, на которой якобы находился Вася (описание представляет собой последовательность реальных высот зданий H_i).

Если фотографию можно увеличить в некоторое **вещественное** количество раз так, чтобы последовательность отмасштабированных высот зданий на фотографии совпала с какой-то непрерывной подпоследовательностью последовательности реальных высот зданий, происходит верификация и Вася получает 1 балл в карму. При этом набор зданий при различных событиях верификации может перекрываться.

Вычислите, какое суммарное количество баллов в карму получит Вася Пупкин после завершения процесса верификации всех его фотографий.

Формат входных данных

Первая строка входных данных содержит одно целое число n — количество фотографий.

Каждая из последующих n строк задаёт одну фотографию. Строка начинается с целого числа $b_i \geq 1$ — количества зданий на фотографии, за которым следуют b_i целых чисел — высоты зданий, перечисленные слева направо.

Последняя строка задаёт описание улицы города и начинается целым числом m — количеством зданий на улице. Далее следуют m ($1 \leq m \leq 3 \cdot 10^5$) целых чисел H_i — высоты зданий, перечисленные слева направо. Сумма всех b_i не превосходит m , все высоты — целые числа между 1 и 10^4 включительно.

Формат выходных данных

Выведите одно целое число — суммарное количество баллов в карму, которое получит Вася после верификации всех фотографий.

Пример

стандартный ввод	стандартный вывод
5 2 20 20 1 42 2 42 84 2 20 10 3 2 4 2 10 5 10 5 10 20 10 5 10 2 1	20