Задача A. Вычислительная биология

Имя входного файла:	`biology.in`
Имя выходного файла:	`biology.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В современной биологии ученым часто приходится иметь дело с последовательностями ДНК. Эти последовательности зачастую являются очень длинными и их ручная обработка требует большого количества времени и сил. Поэтому возникает идея автоматизировать этот процесс.

Для этого можно применять компьютерные методы обработки данных, например, весьма полезными оказываются алгоритмы на строках. В этой задаче последовательность ДНК будет представляться в виде строки, все символы которой входят в множество $\{A, G, C, T\}$.

Пусть даны две последовательности ДНК: $s = s_1 s_2 \ldots s_n$ и $t = t_1 t_2 \ldots t_m$. Будем говорить, что t может получится *в результате эволюции* из s, если s является подпоследовательностью t, то есть существует такая последовательность индексов $1 \le i_1 < i_2 < \ldots < i_m \le m$, что $s_1 = t_{i_1}$, $s_2 = t_{i_2}$, $\ldots$, $s_m = t_{i_m}$. Необходимо выяснить, может ли последовательность t получится в результате эволюции из s.

Формат входного файла

Первая строка входного файла содержит последовательность s, вторая — последовательность t. Размер входного файла не превосходит 256 килобайт.

Формат выходного файла

В выходной файл выведите слово `YES`, если последовательность t могла получится в результате эволюции из s, и слово `NO` — иначе.

Примеры

biology.in	biology.out
GTA AGCTA	YES
AAAG GAAAAAT	NO

Задача B. Скобочки

Имя входного файла:	`brackets.in`
Имя выходного файла:	`brackets.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Напомним, что называется правильной скобочной последовательностью:

- пустая строка является правильной скобочной последовательностью;
- если строка a — правильная скобочная последовательность, то строки (a), $[a]$ — тоже правильные скобочные последовательности;
- если строки a и b — правильные скобочные последовательности, то строка ab — тоже правильная скобочная последовательность.

Задана строка S, состоящая из квадратных и круглых скобок. Разрешается заменять квадратную открывающую скобку (`[`) на круглую открывающую (`(`) и наоборот, а также квадратную закрывающую скобку (`]`) на круглую закрывающую (`)`) и наоборот.

За одно действие разрешается изменить ровно один символ строки. Необходимо за минимальное число действий преобразовать S в правильную скобочную последовательность.

Формат входного файла

Входной файл содержит строку S. Ее длина не превосходит 100 000 символов.

Формат выходного файла

В выходной файл выведите искомое минимальное число действий или -1, если преобразовать S в правильную скобочную последовательность невозможно.

Примеры

brackets.in	brackets.out
`(())[]`	0
`[(])`	2
`(([]]`	-1

Задача C. Количество делителей — 2

Имя входного файла:	`divcnt2.in`
Имя выходного файла:	`divcnt2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Пусть x — натуральное число. Назовем y его *делителем*, если $1 \le y \le x$ и остаток от деления x на y равен нулю.

Задано число x. Найдите количество его делителей, делящихся на каждое из простых чисел, на которое делится x.

Формат входного файла

Первая строка входного файла содержит целое число x ($1 \le x \le 10^{18}$). Все простые делители числа x не превосходят тысячу.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

divcnt2.in	divcnt2.out
12	2
239	1

Задача D. Игра с монеткой

Имя входного файла:	`game.in`
Имя выходного файла:	`game.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Петя играет в интересную игру. Для этой игры необходима монетка. Петя подбрасывает ее n раз и считает, сколько раз выпадает «решка». Если решка выпадает хотя бы m раз, то Петя считает, что он выиграл игру.

Однажды Петя задумался, какова вероятность того, что он выиграет игру. Для этого он хочет найти количество последовательностей результатов подбрасывания монетки, содержащих ровно n подбрасываний, при которых «решка» выпала хотя бы m раз.

Помогите Пете — найдите это число, считая, что при каждом броске монетка может выпасть либо «орлом», либо «решкой».

Формат входного файла

Входной файл содержит два целых числа: n и m ($1 \le n \le 20$, $0 \le m \le n$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

game.in	game.out
2 0	4
3 2	4

Задача E. Задача на движение

Имя входного файла:	`motion.in`
Имя выходного файла:	`motion.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Автомобиль двигался по дороге со скоростью $V =$ _ _ _ _ _ _ *м/с в течение* $T =$ _ _ _ _ _ *секунд и проехал* $S =$ _ _ _ _ _ *метров.*

Некоторые значения уже вписаны, напишите программу, которая будет вписывать все неизвестные значения, чтобы получалось верное утверждение.

Формат входного файла

Во входном файле записаны три целых числа: V, T, S. Известные значения положительны и не превышают 1000. Для обозначения неизвестных значений используется -1.

Формат выходного файла

Если можно вписать значения, чтобы получилось верное утверждение, выдайте тройку положительных чисел V, T, S, иначе выведите `-1 -1 -1`. Все числа в вашем ответе должны отличаться от правильных не более чем на 0.001.

Пример

motion.in	motion.out
`10 5 -1`	`10 5 50`
`10 5 51`	`-1 -1 -1`

Задача F. Прямоугольники

Имя входного файла:	`rect.in`
Имя выходного файла:	`rect.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Найдите количество невырожденных прямоугольников со сторонами, параллельными осям координат, вершины которых лежат в точках с целыми координатами внутри или на границе прямоугольника, противоположные углы которого находятся в точках $(0, 0)$ и (W, H).

Формат входного файла

Входной файл содержит два числа: W и H $(1 \le W, H \le 10^3)$.

Формат выходного файла

В выходной файл выведите ответ на поставленную задачу.

Примеры

rect.in	rect.out
1 1	1
2 1	3

Задача G. Рунные слова

Имя входного файла:	`runes.in`
Имя выходного файла:	`runes.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Руны — это древние магические знаки, которые наши предки использовали как буквы. Говорят, что рунные знаки обладают магическими свойствами, а при сложении рун в слова их магическая сила многократно возрастает. Если кузнец изготовит доспехи и начертит там определенные руны в определенном порядке, то доспехи будут наделены необычайными магическими силами.

Для того, чтобы стать обладателем таких доспехов достаточно просто принести кузнецу начертания этих рунных знаков. А вот, чтобы стать обладателем рунного знака приходилось немало потрудиться. Воины добывали начертания рун других языков и наречий в боях или получали их в качестве наград в благодарность за оказанные услуги.

Но так или иначе и в этом деле развелись жулики. По подозрениям ученых кузнец Игнатус Мошеникус изготавливал благородным войнам фальшивые рунные слова. Из древних преданий ученым стало достоверно известно, что каждая руна записывается из двух, трех или четырех латинских букв. Причем первая буква рунного слова всегда записывается как заглавная, а все остальные являются маленькими. Ученые перевели несколько, выкованных этим кузнецом, рунных слов на латинский язык и теперь нуждаются в Вашей помощи. Проверьте, является ли приведенное слово рунным.

Формат входного файла

В единственной строке входного файла содержится слово. Оно представляет собой непустую строку, длиной не более 100000 символов, содержащую только большие и маленькие буквы латинского алфавита.

Формат выходного файла

Если введенное слово является рунным, запишите в выходной файл слово «`Yes`», в противном случае — «`No`»

Примеры

runes.in	runes.out
`IoTheBest`	`Yes`
`IoItIsWaste`	`No`

Задача H. Подмассив массива

Имя входного файла:	`subarray.in`
Имя выходного файла:	`subarray.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Пусть задан массив целых чисел a_1, a_2, ..., a_n. Назовем его подмассивом $f(i, j)$ массив составленный из чисел $a_i, a_{i+1}, \ldots, a_{j-1}, a_j$. Напишите программу, которая будет выводить подмассивы массива a.

Формат входного файла

Первая строка входного файла содержит число n ($1 \le n \le 1000$) — количество элементов в массиве a. Во второй строке содержатся числа a_1, a_2, ..., a_n разделенные пробелом. Все a_i находятся в диапазоне от -2^{31} до $2^{31}-1$. В третьей строке находится m ($1 \le m \le 100$) — количество подмассивов, которые необходимо вывести. Следующие m строк содержат пары чисел i_k, j_k ($1 \le i_k \le j_k \le n$).

Формат выходного файла

Для каждой пары (i_k, j_k) в отдельной строке выведите подмассив $f(i_k, j_k)$.

Примеры

subarray.in	subarray.out
6 1 2 3 4 5 6 5 1 1 2 6 3 4 5 6 2 4	1 2 3 4 5 6 3 4 5 6 2 3 4