Задача A. Скобочные последовательности

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.

Пусть символы '(', ')', '[', ']' некоторым образом упорядочены. Рассмотрим все правильные скобочные последовательности длины n, начинающиеся с заданной строки s. Требуется найти среди них минимальную в лексикографическом порядке.

Напомним, что строка a лексикографически меньше строки b, если существует такое i, что для любого $j < i$ выполняется равенство $a_j = b_j$ и при этом $a_i < b_i$.

Порядок скобок задается строкой w, состоящей из четырех символов. При этом $w_1 < w_2 < w_3 < w_4$. Например, если $w =$ '()[]', то '(' < ')' < '[' < ']'.

Формат входного файла

Входной файл состоит из трех строк. В первой строке записано натуральное число n ($1 \le n \le 10^5$). Во второй и третьей — строки w и s соответственно. Длина строки s не превосходит n.

Формат выходного файла

В выходной файл выведите наименьшую в лексикографическом порядке правильную скобочную последовательность длины n, начинающуюся со строки s. Гарантируется, что такая существует.

Примеры

brackets.in	brackets.out
6 ()[] ([(	([()])
6][)(([	([][])
4 (][) ()[]	()[]

Задача B. Слова

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

Два слова a и b называются *похожими*, если $a = b$ или $a = b^r$, где b^r получается из b записью символов b в обратном порядке. Например, слова 'abcde' и 'edcba' похожи, а слова 'aba' и 'abb' нет.

Требуется посчитать максимально возможное число попарно непохожих слов длины n над алфавитом из k символов.

Например, при $n = 3$ и $k = 2$ есть максимум шесть попарно непохожих слов: 'aaa', 'aab', 'aba', 'abb', 'bab', 'bbb'.

Формат входного файла

В первой строке записаны два натуральных числа n ($1 \le n \le 10^9$) и k ($1 \le k \le 10^9$).

Формат выходного файла

В выходной файл выведите остаток от деления искомого количества на $10^9 + 7$.

Примеры

words.in	words.out
3 2	6

Задача C. Преобразование отрезков

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

Петя работает в лаборатории Прикладных Исследований Лингвистических Абстракций (ПИЛА). Его последняя работа связана с исследованиями преобразований строк, он исследует так называемые *преобразования отрезков*.

Преобразование отрезка, обозначаемое как $T(i, j, k)$, примененное к строке s, состоящей из символов, встречающихся в описаниях ДНК ($\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$) осуществляется следующим образом: символы s, находящиеся на позициях между i и j, включительно, заменяются на символы, которые находятся на k позиций позже в последовательности `A`, `C`, `G`, `T` ($1 \le k \le 3$). При этом последовательность считается зацикленной, то есть после `T` идет `A`. Например, послед применения преобразования $T(2, 5, 2)$ к строке “`AGGTCAT`” получается строка “`AAACTAT`”.

По заданным двум строкам s и t равной длины, выясните, какое минимальное количество преобразований отрезка требуется применить к строке s, чтобы она оказалась равна t.

Формат входного файла

Первая строка входного файла содержит s. Вторая строка содержит t. Обе строки имеют одну и ту же длину, не превышающую 100 и содержат только символы из множества $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$.

Формат выходного файла

Первая строка должна содержать k — минимальное количество преобразований отрезка, которое требуется, чтобы получить из s строку t. Следующие k строк должны содержать по три целых числа и описывать эти преобразования.

Пример

transform.in	transform.out
AGGTCAT AAACTAA	2 2 5 2 7 7 1

Задача D. Минимумы на отрезках

Имя входного файла:	`minima.in`
Имя выходного файла:	`minima.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайта

Вам задан массив $x[1 \ldots n]$ и число m. Для всех i от 1 до $n-m+1$ найдите минимум среди чисел $x[i], x[i+1], \ldots, x[i+m-1]$ и верните сумму этих минимумов.

Формат входного файла

Первая строка входного файла содержит три целых числа n, m и k ($1 \le n \le 30\,000\,000$, $1 \le m \le n$, $min(n, 2) \le k \le \min(n, 1000)$). Вторая строка содержит три целых числа a, b и c ($-2^{31} \le a, b, c \le 2^{31}-1$). Третья строка содержит k целых чисел: $x[1], x[2], \ldots, x[k]$ ($-2^{31} \le x[i] \le 2^{31}-1$).

Остальные элементы массива могут быть вычислены по формуле: $x[i] = f(a \cdot x[i-2] + b \cdot x[i-1] + c)$. Здесь $f(y)$ возвращает такое число z, где $-2^{31} \le z \le 2^{31}-1$, что $y - z$ делится без остатка на 2^{32}.

Формат выходного файла

Выведите одно целое число — сумму минимумов всех подмассивов длины m заданного массива.

Пример

minima.in	minima.out
10 3 2 1 1 0 0 1	33
1000000 15 5 283471207 23947205 3 17625384 939393931 1838388 912740247 290470294	-1879262596173354