

## Задача А. Кувшинки

Имя входного файла: `parenth2.in`  
Имя выходного файла: `parenth2.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

*Правильной скобочной последовательностью из двух типов скобок из  $2n$  скобок называется такая последовательность круглых и квадратных скобок, которая может встречаться в некотором арифметическом выражении. Например,  $() [] ()$  и  $((()))$  являются правильными скобочными последовательностями из двух типов скобок, а  $((())) [$  и  $[] ] [$  — нет.*

Все правильные скобочные последовательности из двух типов скобок можно упорядочить в лексикографическом порядке, считая, что порядок скобок соответствует их кодам символов:  $(' < ') < '[' < ']$ . Скажем, при  $n = 2$  список упорядоченных правильных скобочных последовательностей из двух типов скобок будет выглядеть так:  $((()))$ ,  $()()$ ,  $() []$ ,  $([])$ ,  $[()]$ ,  $[[]]$ ,  $[]()$ ,  $[] []$ .

В этой задаче требуется найти правильную скобочную последовательность из двух типов скобок по лексикографическому номеру (нумерация ведется с нуля).

### Формат входного файла

Два числа  $n$  и  $x$  ( $1 \leq n \leq 20$ ),  $x$  задаёт номер существующей правильной скобочной последовательности из двух типов скобок.

### Формат выходного файла

Выведите строку из  $2n$  круглых и квадратных скобок, задающих требуемую правильную скобочную последовательность из двух типов скобок.

### Примеры

	<code>parenth2.in</code>	<code>parenth2.out</code>
2 1		$()()$
2 4		$[()]$

## Задача В. Range Variation Query

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

В начальный момент времени последовательность  $a_n$  задана следующей формулой:  $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$ .

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов  $a_i, a_{i+1}, \dots, a_j$ ;
- присвоить элементу  $a_i$  значение  $j$ .

### Формат входного файла

Первая строка входного файла содержит натуральное число  $k$  — количество запросов ( $1 \leq k \leq 100\,000$ ). Следующие  $k$  строк содержат запросы, по одному на строке. Запрос номер  $i$  описывается двумя целыми числами  $x_i, y_i$ .

Если  $x_i > 0$ , то требуется найти разность между максимальным и минимальным значениями среди элементов  $a_{x_i}, \dots, a_{y_i}$ . При этом  $1 \leq x_i \leq y_i \leq 100\,000$ .

Если  $x_i < 0$ , то требуется присвоить элементу  $a_{|x_i|}$  значение  $y_i$ . В этом случае  $-100\,000 \leq x_i \leq -1$  и  $|y_i| \leq 100\,000$ .

### Формат выходного файла

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

### Примеры

<code>rvq.in</code>	<code>rvq.out</code>
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

## Задача С. Задобрите Инти

Имя входного файла: `makeintihappy.in`  
Имя выходного файла: `makeintihappy.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Как вы помните, племя Майя вымерло, племя Йюля вынуждено было распасться и покинуть обжитые территории, и вот теперь племени Августя тоже угрожает опасность. Задоблив бога дождя Тлалока, они напрочь забыли про бога солнца Инти, который будучи обделённым вниманием грозиться ниспослать испепеляющую жару, если жители племени не порадуют его божественный взор новой коллекцией симпатичных узоров, выложенных из огромных гранитных плит, покрашенных в чёрный и белый цвета. Напомним, что симпатичным узором называется прямоугольник шириной  $W$  и высотой  $H$  ни какой квадрат  $2$  на  $2$  которого не покрашен в один цвет.

В данной задаче таскать гранитные плиты Вам не придётся, однако потребуется определить  $K$ -й лексикографически симпатичный узор. Один узор будем считать лексикографически меньше другого, если при просмотре узоров по столбцам сверху вниз найдётся такая позиция, что все плиты рассмотренные ранее совпадают попарно по цвету, а плита на данной позиции у первого узора белая, а у второго — чёрная.

### Формат входного файла

Во входном файле заданы числа  $W$ ,  $H$  и  $K$ . ( $1 \leq W \leq 1000$ ,  $1 \leq H \leq 10$ ,  $1 \leq K \leq 10^{18}$ )

### Формат выходного файла

Выведите требуемый симпатичный узор, обозначая белые плиты символом  $'w'$ , а чёрные — символом  $'b'$ . Если не существует узора с таким номером — выведите "Impossible" (без кавычек).

### Примеры

<code>makeintihappy.in</code>	<code>makeintihappy.out</code>
5 1 17	bwww
7 4 1234567	wbwbwb wbbwww wwwbbwb wbwbwb

## Задача D. Дерево отрезков с операцией НОД

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

Реализуйте структуру данных для эффективного вычисления НОД нескольких подряд идущих элементов массива.

### Формат входного файла

В первой строке вводится одно натуральное число  $N$  ( $1 \leq N \leq 100\,000$ ) — количество чисел в массиве.

Во второй строке вводятся  $N$  чисел от 1 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число  $K$  ( $1 \leq K \leq 30\,000$ ) — количество запросов на вычисление НОД.

В следующих  $K$  строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

### Формат выходного файла

Для каждого запроса выведите НОД всех чисел соответствующего участка массива. Числа выводите в одну строку через пробел.

### Примеры

<code>segment-gcd.in</code>	<code>segment-gcd.out</code>
5	2
2 2 2 1 5	1
2	
2 3	
2 5	

## Задача Е. Длинные домино

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

Найдите количество способов замостить прямоугольник  $m \times n$  доминошками размерами  $3 \times 1$ . Каждая доминошка должна лежать полностью внутри прямоугольника. Никакие две доминошки не должны накладываться друг на друга. Доминошки могут быть ориентированы как горизонтально, так и вертикально.

### Формат входного файла

В единственной строке входного файла даны два целых числа  $m$  и  $n$  ( $1 \leq m \leq 9, 1 \leq n \leq 30$ ).

### Формат выходного файла

В выходной файл выведите единственное число — количество способов замостить прямоугольник  $n \times m$  доминошками.

### Примеры

<code>longdominoes.in</code>	<code>longdominoes.out</code>
3 3	2
3 10	28

## Задача F. Фенечка

Имя входного файла: `bracelet.in`  
Имя выходного файла: `bracelet.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Саша находится в процессе творческого поиска. Она хочет сплести ещё одну фенечку, но испытывает сложности при выборе цветов. Сейчас все  $n$  ниток, которые она планирует использовать для плетения, выложены в ряд. В процессе размышления Саша время от времени заменяет нитку одного цвета ниткой другого, а также для проверки того, что узор получается тем, который подразумевается, проверяет, что некоторые последовательности цветов ниток равны.

Напишите программу, которая автоматизирует эти проверки.

### Формат входного файла

В первой строке входного файла записаны два целых числа  $n$  и  $k$  — количество ниток в фенечке и запросов к программе, соответственно.  $1 \leq n, k \leq 100\,000$ . Во второй строке записана строка из  $n$  символов — цвета ниток в начальном состоянии. Каждый цвет обозначается строчной или прописной буквой латинского алфавита или цифрой.

В следующих  $k$  строках заданы запросы двух видов:

- «\*  $i$   $c$ » — заменить нитку с номером  $i$  на нитку цвета  $c$ ,
- «?  $i$   $j$   $len$ » — проверить, равны ли последовательности цветов ниток, начинающиеся в позициях  $i$  и  $j$  и имеющие длину  $len$ .

### Формат выходного файла

Для каждого запроса второго вида выведите «+», если последовательности равны, или «-» в противном случае.

### Примеры

bracelet.in	bracelet.out
7 4 abacaba ? 1 5 3 * 6 c ? 2 6 2 ? 3 5 3	+ - +
10 5 vznwsempqf * 8 c * 2 z ? 5 4 5 * 6 p ? 5 4 1	- -