Задача C. Предок

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

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

Формат входного файла

Первая строка входного файла содержит натуральное число n ($1 \leqslant n \leqslant 100\,000$) — количество вершин в дереве. Во второй строке находится n чисел, i-ое из которых определяет номер непосредственного родителя вершины с номером i. Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится число m ($1 \leqslant m \leqslant 100\,000$) — количество запросов. Каждая из следующих m строк содержит два различных числа a и b ($1 \leqslant a, b \leqslant n$).

Формат выходного файла

Для каждого из m запросов выведите на отдельной строке число 1, если вершина a является одним из предков вершины b, и 0 в противном случае.

Примеры

ancestor.in	ancestor.out
6 0 1 1 2 3 3 5 4 1 1 4 3 6 2 6 6 5	0 1 1 0 0

Distance. Расстояние между вершинами

Имя входного файла: `distance.in`
Имя выходного файла: `distance.out`

Коль Дейкстру́ писать без кучи,
То тайм-лимит ты получишь...
А в совсем крутой задаче
Юзай кучу Фибоначчи!

Спектакль преподавателей ЛКШ.июль-2007

Дан взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \leqslant s, t \leqslant n$, $s \neq t$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \leqslant b_i, e_i \leqslant n$, $0 \leqslant w_i \leqslant 100$).

$n \leqslant 100\,000$, $m \leqslant 200\,000$.

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t.

Если путь из s в t не существует, выведите -1.

Пример

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

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- `cut` — разрезать граф, то есть удалить из него ребро;
- `ask` — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа `cut` рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа `ask`.

Формат входного файла

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа n, количество рёбер m и количество операций k ($1 \leqslant n \leqslant 50\,000$, $0 \leqslant m \leqslant 100\,000$, $m \leqslant k \leqslant 150\,000$).

Следующие m строк задают рёбра графа; i-я из этих строк содержит два числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$), разделённые пробелами, — номера концов i-го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют k строк, описывающих операции. Операция типа `cut` задаётся строкой «`cut` u v» ($1 \leqslant u, v \leqslant n$), которая означает, что из графа удаляют ребро между вершинами u и v. Операция типа `ask` задаётся строкой «`ask` u v» ($1 \leqslant u, v \leqslant n$), которая означает, что необходимо узнать, лежат ли в данный момент вершины u и v в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа `cut` ровно один раз.

Формат выходного файла

Для каждой операции `ask` во входном файле выведите на отдельной строке слово «`YES`», если две указанные вершины лежат в одной компоненте связности, и «`NO`» в противном случае. Порядок ответов должен соответствовать порядку операций `ask` во входном файле.

Пример

cutting.in	cutting.out
3 3 7 1 2 2 3 3 1 ask 3 3 cut 1 2 ask 1 2 cut 1 3 ask 2 1 cut 2 3 ask 3 1	YES YES NO NO

Задача D. Коровы - в стойла

Имя входного файла:	`cows.in`
Имя выходного файла:	`cows.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

На прямой расположены стойла, в которые необходимо расставить коров так, чтобы минимальное растояние между коровами было как можно больше.

Формат входного файла

В первой строке вводятся числа N ($2 < N < 10\,001$) — количество стойл и K ($1 < K < N$) — количество коров. Во второй строке задаются N натуральных чисел в порядке возрастания — координаты стойл (координаты не превосходят 10^9).

Формат выходного файла

Выведите одно число — наибольшее возможное допустимое расстояние.

Примеры

cows.in	cows.out
5 3 1 2 3 100 1000	99

Задача B. Хипуй!

Имя входного файла:	`heap.in`
Имя выходного файла:	`heap.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

В этой задаче вам необходимо организовать структуру данных Heap для хранения целых чисел, над которой определены следующие операции:

- `Insert(X)` — добавить в Heap число X;
- `Extract` — достать из Heap наибольшее число (удалив его при этом).

Формат входного файла

Во входном файле записано количество команд N ($1 \leqslant N \leqslant 100\,000$), потом последовательность из N команд, каждая в своей строке.

Каждая команда имеет такой формат: „0 `<число>`“ или „1“, что означает соответственно операции `Insert(<число>)` и `Extract`. Добавляемые числа находятся в интервале от 1 до 10^7 включительно.

Гарантируется, что при выполенении команды `Extract` в структуре находится по крайней мере один элемент.

Формат выходного файла

В выходной файл для каждой команды извлечения необходимо вывести число, полученное при выполнении команды `Extract`.

Примеры

heap.in	heap.out
7 0 100 0 10 1 0 5 0 30 0 50 1	100 50

Три единицы подряд

Имя входного файла:	`ones.in`
Имя выходного файла:	`ones.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

По данному числу N определите количество последовательностей из нулей и единиц длины N, в которых никакие три единицы не стоят рядом.

Формат входного файла

Во входном файле написано натуральное число N, не превосходящее 35.

Формат выходного файла

Выведите количество искомых последовательностей. Гарантируется, что ответ не превосходит
$2^{31} - 1$.

Примеры

ones.in	ones.out
4	13