

Упражнение «Произвольное дерево»

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени: 1 секунда на тест

Дерево задано множеством пар "предок - потомок". Найти вершину(ы) дерева наибольшей степени, количество листьев в дереве, высоту дерева, количество вершин на каждом уровне.

Вход

Во входном файле записано количество вершин дерева N ($1 \leq N \leq 10000$) и $N-1$ пара чисел (P , C), где P - номер вершины - предка, C - номер вершины - потомка. Вершины пронумерованы от 1 до N . Пары (P , C) записаны в произвольном порядке.

Выход

Запишите в выходной файл:

- наибольшую степень вершины;
- номера вершин наибольшей степени;
- количество листьев;
- высоту дерева;
- количество вершин на каждом уровне, начиная с корня.

Примеры входа и выхода

input.txt	output.txt
10 9 10 2 9 9 8 4 7 4 6 4 5 2 4 2 3 9 1	3 2 4 9 7 3 1 3 6

Упражнение «Бинарное дерево поиска»

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени: 1 секунда на тест

Дана последовательность операций «добавить вершину», «удалить вершину» и «найти ключ» над бинарным деревом поиска. Выполнить эти операции.

Вход

Во входном файле записана последовательность команд вида “+**s**” или “-**s**” или “?**s**” или “**Е**”. Здесь **s** - слово длиной не более 10 символов. Команда “+**s**” означает добавить в дерево слово **s**, команда “-**s**” означает удалить из дерева слово **s**, команда “?**s**” - найти в дереве слово **s**. Команда “**Е**” означает завершить операции с деревом. Каждая команда записана в отдельной строке, начиная с первой позиции, и не содержит пробелов.

Выход

В выходной файл запишите результаты выполнения команд “?**s**” в том порядке, в котором они присутствуют во входном файле. Если слово найдено, записать в файл слово и его глубину в дереве (через пробел), в противном случае – слово и число ноль (через пробел).

Примеры входа и выхода

input.txt	output.txt
+глокая	бокра 0
+кузdra	кузdra 0
?бокpa	глокая 1
-кузdra	
?кузdra	
?глокая	
Е	

Упражнение «Декартово дерево»

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени: 1 секунда на тест

Выполнить последовательность операций «добавить вершину», «удалить вершину» и «найти ключ» над декартовым деревом.

Вход

Во входном файле записана последовательность команд вида “+**x y**” или “-**x**” или “?**x**” или “**E**”. Здесь **x, y** - натуральные числа, не превосходящие 10^6 . Команда “+**x y**” означает добавить в декартово дерево вершину с ключами **x, y**, команда “-**x**” означает удалить из декартова дерева вершину с ключом **x**, команда “?**x**” - найти в декартовом дереве вершину с ключом **x**. Команда “**E**” означает завершить операции с декартовым деревом. Каждая команда записана в отдельной строке, начиная с первой позиции, и не содержит пробелов, кроме команды “+”, содержащей один пробел.

Выход

В выходной файл записать результаты выполнения команд “?**x**” в том порядке, в котором они присутствуют во входном файле. Если вершина найдена, записать в файл **x-ключ** и **глубину вершины в дереве** (глубина корня равна 1), в противном случае - **x-ключ** и **число ноль**.

Примеры входа и выхода

input.txt	output.txt
?1 +3 6 -1 +2 1 -8 +6 4 -2 ?6 -3 -5 E	1 0 6 2

Упражнение «АВЛ-дерево»

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени: 1 секунда на тест

Дана последовательность операций «добавить вершину», «удалить вершину» и «найти ключ» над АВЛ-деревом. Выполнить эти операции.

Вход

Во входном файле записана последовательность команд вида “+**s**” или “-**s**” или “?**s**” или “**Е**”. Здесь **s** - слово длиной не более 10 символов. Команда “+**s**” означает добавить в дерево слово **s**, команда “-**s**” означает удалить из дерева слово **s**, команда “?**s**” - найти в дереве слово **s**. Команда “**Е**” означает завершить операции с деревом. Каждая команда записана в отдельной строке, начиная с первой позиции, и не содержит пробелов.

Выход

В выходной файл запишите результаты выполнения команд “?**s**” в том порядке, в котором они присутствуют во входном файле. Если слово найдено, записать в файл слово и его глубину в дереве (через пробел), в противном случае – слово и число ноль (через пробел).

Примеры входа и выхода

input.txt	output.txt
+глокая	бокра 0
+куздра	куздра 0
?бокра	глокая 1
-куздра	
?куздра	
?глокая	
Е	

Упражнение «SPLAY-дерево»

Входной файл: **input.txt**

Выходной файл: **output.txt**

Ограничение времени: **1 секунда на тест**

Дана последовательность операций «добавить вершину», «удалить вершину» и «найти ключ» над SPLAY-деревом. Выполнить эти операции.

Вход

Во входном файле записана последовательность команд вида “**+s**” или “**-s**” или “**?s**” или “**E**”. Здесь **s** - слово длиной не более 10 символов. Команда “**+s**” означает добавить в дерево слово **s**, команда “**-s**” означает удалить из дерева слово **s**, команда **?s** - найти в дереве слово **s**. Команда “**E**” означает завершить операции с деревом. Каждая команда записана в отдельной строке, начиная с первой позиции, и не содержит пробелов.

Выход

В выходной файл запишите результаты выполнения команд “**?s**” в том порядке, в котором они присутствуют во входном файле. Если слово найдено, записать в файл слово и его глубину в дереве (через пробел), в противном случае – слово и число ноль (через пробел).

Примеры входа и выхода

input.txt	output.txt
+глокая +куздра ?бокра -куздра ?куздра ?глокая E	бокра 0 куздра 0 глокая 1

Упражнение «Декартово дерево по неявному ключу»

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени: 1 секунда на тест

Дана последовательность операций «добавить вершину», «вставить вершину», «удалить вершины» и «найти ключ» над декартовым деревом по неявному ключу. Выполнить эти операции.

Вход

Во входном файле записана последовательность команд вида **"ADD s"** или **"INS s k"** или **"DEL a b"** или **"KEY p"**, или **"END"**. Здесь *s* - слово длиной не более 10 символов. Команда **"ADD s"** означает "добавить в дерево справа вершину со словом *s*". Команда **"INS s k"** означает «вставить в дерево вершину со словом *s* в *k*-ю позицию». Команда **"DEL a b"** означает "удалить из вершины с номерами от *a* до *b* включительно. Команда **"KEY p"** означает "вывести слово, записанное в вершине с номером *p*". Команда **"END"** означает "завершить операции с деревом". Здесь *k*, *a*, *b*, *p* – натуральные числа. Каждая команда записана в отдельной строке, начиная с первой позиции, слова и числа, составляющие команду, разделяются ровно одним пробелом. Вершины дерева нумеруются, начиная с единицы. Если в команде **"INS s k"** величина *k* превышает текущее количество вершин, вместо неё следует выполнить команду **"ADD s"**. Если в команде **"DEL a b"** величина *a* превышает текущее количество вершин, то ничего не нужно делать, в противном случае, если величина *b* превышает текущее количество вершин, то следует удалить все вершины, начиная с *a*-ой.

Выход

В выходной файл запишите результаты выполнения команд **"KEY p"** в том порядке, в котором они присутствуют во входном файле. Если величина *p* не превышает текущее количество вершин, то запишите в отдельной строке слово, записанное в вершине с номером *p*, в противном случае запишите в отдельной строке **«OUT OF RANGE»**.

Примеры входа и выхода

input.txt	output.txt
ADD глокая	куздра
ADD куздра	OUT OF RANGE
INS 1 бокра	бокра
KEY 3	
DEL 2 5	
KEY 2	
KEY 1	
END	

Задача «Река»

Входной файл: river.in

Выходной файл: river.out

Ограничение времени: 1 секунда на тест

Во Флатландии протекает богатая рыбой река Большой Флат. Много лет назад река была поделена между n рыболовными предприятиями, каждое из которых получило непрерывный отрезок реки. При этом i -е предприятие, если рассматривать их по порядку, начиная от истока, изначально получило отрезок реки длиной a_i . С тех пор с рыболовными предприятиями во Флатландии k раз происходили различные события. Каждое из событий было одного из двух типов: банкротство некоторого предприятия или разделение некоторого предприятия на два. При некоторых событиях отрезок реки, принадлежащий предприятию, с которым это событие происходит, делится на две части. Каждый такой отрезок имеет длину большую или равную 2. Деление происходит по следующему правилу. Если отрезок имеет четную длину, то он делится на две равные части. Иначе он делится на две части, длины которых различаются ровно на единицу, при этом часть, которая ближе к истоку реки, имеет меньшую длину. При банкротстве предприятия происходит следующее. Отрезок реки, принадлежавший обанкротившемуся предприятию, переходит к его соседям. Если у обанкротившегося предприятия один сосед, то этому соседу целиком передается отрезок реки обанкротившегося предприятия. Если же соседей двое, то отрезок реки делится на две части описанным выше способом, после чего каждый из соседей присоединяет к своему отрезку ближайшую к нему часть. При разделении предприятия отрезок реки, принадлежавший разделяемому предприятию, всегда делится на две части описанным выше способом. Разделившееся предприятие ликвидируется, и образуются два новых предприятия. Таким образом, после каждого события каждое предприятие владеет некоторым отрезком реки. Министерство финансов Флатландии предлагает ввести налог на рыболовные предприятия, пропорциональный квадрату длины отрезка реки, принадлежащего соответствующему предприятию. Чтобы проанализировать, как будет работать этот налог, министр хочет по имеющимся данным узнать, как изменялась величина, равная сумме квадратов длин отрезков реки, принадлежащих предприятиям, после каждого произошедшего события. Требуется написать программу, которая по заданному начальному разделению реки между предприятиями и списку событий, происходивших с предприятиями, определит, чему равна сумма квадратов длин отрезков реки, принадлежащих предприятиям, в начальный момент времени и после каждого события.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и p — исходное количество предприятий ($2 \leq n \leq 100\,000$). Вторая строка входного файла содержит n целых чисел a_1, a_2, \dots, a_n — длины исходных отрезков реки. Третья строка входного файла содержит целое число k — количество событий, происходивших с предприятиями ($1 \leq k \leq 100\,000$). Последующие k строк содержат описания событий, i -я строка содержит два целых числа: e_i и v_i — тип события и номер предприятия, с которым оно произошло. Значение $e_i = 1$ означает, что предприятие, которое после всех предыдущих событий является v_i -м по порядку, если считать с единицы от истока реки, обанкротилось, а значение $e_i = 2$ означает, что это предприятие разделилось на два. Гарантируется, что значение v_i не превышает текущее количество предприятий. Гарантируется, что если отрезок предприятия при банкротстве или разделении требуется поделить на две части, то он имеет длину большую или равную 2. Гарантируется, что если на реке осталось единственное предприятие, оно не банкротится.

Формат выходного файла

Выходной файл должен содержать $(k + 1)$ целых чисел, по одному в строке. Первая строка должна содержать исходную сумму квадратов длин отрезков реки, а каждая из последующих k строк — сумму квадратов длин отрезков реки после очередного события.

Пример входных и выходных файлов

river.in	river.out
4 0	75
3 5 5 4	105
5	73
1 1	101
2 1	83
1 3	113
2 2	
1 3	