A. Расстояние от корня

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

На вход подаётся корневое дерево. В нём нужно найти все вершины, максимально удалённые от корня. Напомним, что расстоянием между вершинами называется число рёбер в кратчайшем пути между ними.

Входные данные

В первой строке приведено число вершин в дереве $1 \leq n \leq 100$. В следующих $n - 1$ строках заданы вершины, являющиеся предками вершин $2, 3, \ldots, n$. Вершина 1 является корнем дерева.

Выходные данные

В первой строке выведите максимальное расстояние от корня до остальных вершин дерева. Во второй строке выведите сколько вершин дерева находятся от корня на таком расстоянии. В третьей строке выведите номера этих вершин через пробел в порядке возрастания.

Пример

входные данные	Скопировать
3 1 1	

выходные данные	Скопировать
1 2 2 3	

B. Обход раскрашенного дерева

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 64 мегабайта

ввод: стандартный ввод

вывод: стандартный вывод

На вход задачи подаётся дерево, каждый узел которого покрашен в белый, чёрный или серый цвет. Необходимо найти сколько вершин на путях от корня к листьям (считая листья) имеют белый цвет, при условии что до попадания в белую вершину путь прошёл через чёрную.

Входные данные

Деревья закодированы скобочными выражениями. После открывающей скобки идёт цвет вершины b (black — чёрный), w (white — белый) или g (gray — серый) и если у узла есть потомки, то потомки перечислены через запятую. Длина описания дерева не превосходит 10000 символов.

Выходные данные

Выведите число N – количество искомых вершин.

Примеры

входные данные	Скопировать
(g,(w),(b,(w),(g,(w))),(b,(w)))	

выходные данные	Скопировать
3	

входные данные	Скопировать
(w,(b),(b,(g,(w),(b))))	

выходные данные	Скопировать
1	

C. Конкатенация и поворот

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 64 мегабайта
ввод: стандартный ввод
вывод: стандартный вывод

Определим операции конкатенации $\cdot$ и разворота R. Конкатенация склеивает два слова в одно: для слов $u = ab$ и $v = abb$ их конкатенация есть $u \cdot v = uv = ababb$. Операция разворота переставляет буквы слова в обратном порядке: $uR = ba$, $vR = bba$, $(uv)R = bbaba$. Заметим, что для любых u и v, $(uv)R = (vR)(uR)$.

Дано дерево, каждый узел которого является операцией конкатенации. К некоторым узлам применена операция разворота – слово, которое окажется вычисленным в узле необходимо развернуть, прежде чем продолжать вычисление. Необходимо вычислить получившееся в корне слово.

Входные данные

Деревья закодированы скобочными выражениями. Если у узла более одного ребёнка, то соответствующие детям поддеревья перечислены через запятую. Если к узлу применяется операция разворота, то она записана после закрывающей скобки. Общая длина описания дерева не превосходит 100 символов. В листьях записаны слова, состоящие из строчных английских букв.

Выходные данные

Выведите слово, получающееся в корне дерева после всех вычислений.

Примеры

входные данные	Скопировать
((ab)R,(abb))	

выходные данные	Скопировать
baabb	

входные данные	Скопировать
((ab)R,(abb))R	

выходные данные	Скопировать
bbaab	

D. Кодирование по Хаффману с кучей

ограничение по времени на тест: 3.0 с

ограничение по памяти на тест: 512 МБ

ввод: standard input

вывод: standard output

Кодирование по Хаффману можно эффективно строить с помощью минимальной кучи — необходимо построить кучу для узлов будущего дерева с ключами-частотами, дальше извлекать по два элемента с минимальной частотой, назначать им в качестве родителя новую вершину и возвращать её в кучу вместо извлечённых элементов.

Реализуйте данный алгоритм для построения кода Хаффмана. В качестве результата выведите стоимость кода. Стоимостью кода называется величина $\sum_{i=1}^{n} f_i \cdot d_i$, где f_i — частота i-го символа, а d_i — его глубина в дереве. Корень имеет глубину 0 (глубина равна длине кода символа).

Входные данные

В первой строке содержится целое число $N \leq 10^6$ — количество частот.

Во второй строке содержится N чисел $f_1, f_2, \ldots, f_N$ — частоты каждого символа; $1 \leq f_i \leq 10^3$.

Выходные данные

Выведите целое число — наименьшую стоимость префиксного кода (стоимость кода Хаффмана) при данных частотах.

Пример

входные данные	Скопировать
4 1 3 4 2	

выходные данные	Скопировать
19	