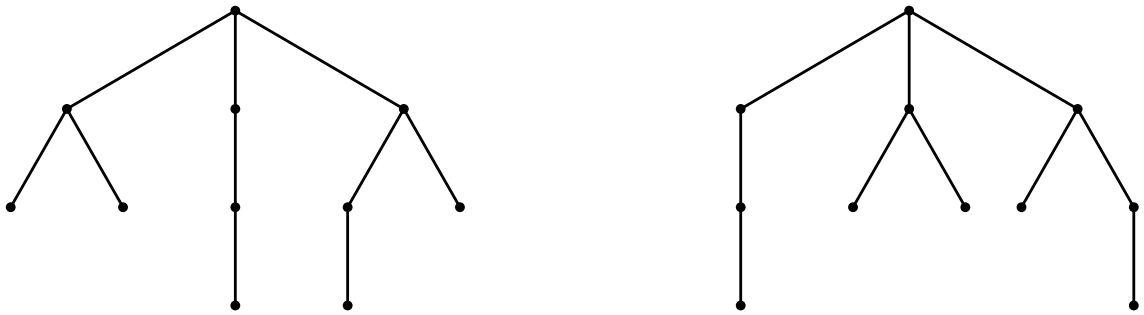
Задача 3. Скобки и деревья

Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Это интерактивная задача с двойным запуском.

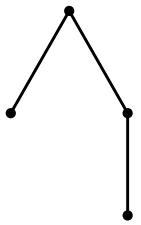
В этой задаче рассматриваются *корневые деревья без порядка на детях*. Корневое дерево без порядка на детях состоит из корня, у которого может быть ноль или более детей. Каждый ребенок в свою очередь является корневым деревом без порядка на детях. При этом, как и следует из названия дерева, порядок, в котором перечисляются дети, не важен, то есть деревья, изображенные на рисунке ниже являются одним и тем же деревом без порядка на детях. Далее будем называть корневые деревья без порядка на детях просто *деревьями*.



Любое дерево можно закодировать в виде *правильной скобочной последовательности* (далее — ПСП) следующим образом:

- Дерево, состоящее из одной вершины, кодируется как «`()`».
- Пусть после удаления корня дерево распадается на поддеревья $t_1, t_2, \ldots, t_k$, где k — количество детей корня исходного дерева. Положим, что $s_1, \ldots, s_k$ — строки, которые кодируют деревья $t_1, \ldots, t_k$. Тогда для любой перестановки $a = [a_1, a_2, ..., a_k]$ чисел от 1 до k исходное дерево может быть закодировано ПСП «`(`$s_{a_1} s_{a_2} \ldots s_{a_k}$`)`».

Обратите внимание, что одно и то же дерево может быть закодировано различными ПСП. Например, дерево, изображенное на рисунке ниже, может быть закодировано с помощью ПСП «`(()(()))`» или «`((())())`».



Вам требуется научиться кодировать произвольную последовательность деревьев $u_1, \ldots, u_n$ в виде одного корневого дерева w. Чтобы проверить, что ваш способ кодирования корректный, ваше решение будет запущено два раза.

Первый запуск

При первом запуске вашей программе на вход подаются n ПСП, каждая из которых представляет собой код s_i некоторого корневого дерева. В ответ ваша программа должна вывести ПСП, которая кодирует произвольное корневое дерево w. В различных подзадачах накладываются различные ограничения на количество вершин в дереве w в зависимости от суммарного количества вершин в исходных деревьях.

Второй запуск

На втором запуске вашей программе подаётся единственная ПСП, которая кодирует дерево w, которое ваша программа вывела при первом запуске. При этом на вход может быть подана любая

подходящая ПСП, которая кодирует дерево w, не обязательно та, которая была выведена вашей программой при первом запуске.

В ответ ваша программа должна вывести ПСП, которые кодируют те же деревья, которые были поданы при первом запуске, в том же порядке. Для каждого дерева вы можете вывести любую кодирующую его ПСП, но порядок самих деревьев в последовательности должен быть таким же, как при первом запуске программы.

Протокол взаимодействия

В начале каждого запуска ваша программа должна прочитать одно число t, равное 1 или 2 — номер запуска.

Первый запуск

При первом запуске необходимо обработать несколько наборов входных данных. Каждый набор подаётся на стандартный поток ввода интерактивно, то есть перед тем, как считать очередной набор, ваша программа должна вывести ответ для всех предыдущих наборов входных данных и сбросить буфер стандартного потока вывода.

В первой строке каждого набора входных данных записано одно число n — количество деревьев, которое нужно закодировать. Если n равно 0, то это означает, что все наборы входных данных обработаны и программа должна завершить работу. Иначе в следующих n строках следуют описания деревьев.

Каждое дерево задается одной строкой s_i, которая состоит из символов «(» и «)» — ПСП, которая кодирует i-е дерево описанным в условии образом. Гарантируется, что s_i задает корректное дерево.

Для данного набора входных данных программа должна вывести ПСП, которая кодирует некоторое дерево w. После вывода дерева необходимо вывести символ конца строки и сбросить буфер потока вывода.

В данной задаче работа программы жюри при первом запуске является адаптивной. Это означает, что программа жюри на первом запуске может использовать выведенные вами в предыдущих наборах входных данных текущего теста деревья w при генерации нового набора входных данных.

Второй запуск

На втором запуске необходимо обработать несколько наборов входных данных. Каждый из наборов входных данных задаётся строкой s. Если строка s равна «0», то вы обработали все наборы входных данных, и программа должна завершить работу. Иначе s содержит некоторую ПСП, кодирующую какое-то дерево w, которое программа построила при первом запуске.

Для каждого такого дерева необходимо вывести в отдельной строке одно число n — количество декодированных деревьев.

В следующей строке требуется вывести n ПСП, которые кодируют в соответствующем порядке те же деревья, что кодировали строки $s_1, \ldots, s_n$, поданные при первом запуске, в одну строку, разделяя их символом «+». Например, если нужно вывести ПСП «(())» и «((()))» в таком порядке, то вывод должен быть таким: в первой строке «2», а во второй строке «(())+((()))».

После вывода числа n и вывода строки с описанием деревьев необходимо перевести строку и сбросить буфер поток вывода.

В каждом из наборов данных во втором запуске вашей программе на вход может подаваться любое из деревьев, полученных при первом запуске.

Замечание

Не забывайте переводить строку после каждого вывода. Обратитесь к памятке участника, чтобы узнать, как правильно сбрасывать поток вывода в интерактивных задачах.

Система оценивания

Обозначим за s суммарную длину ПСП в одном наборе входных данных, а за m — размер вывода вашей программы на первом запуске для этого набора входных данных. Для каждой подзадачи определена функция $f(x)$. Подзадача считается пройденной, если для каждого набора входных данных выполняется $m \leqslant f(s)$, а также если все деревья были корректно восстановлены.

Обозначим за t_i количество вершин в i-м дереве. Тогда длина строки s_i равна $2t_i$.

Также гарантируется, что сумма s по всем наборам входных данных одного теста не превосходит 10^6, а количество наборов входных данных в каждом тесте не превосходит 100.

Подзадача	Баллы	$f(x)$	Дополнительные ограничения		Необх. подз.
			s	Дополнительно	
1	13	$f(x) = x + 2000$	$s \leqslant 200\,000$	При втором запуске даются ПСП, точно совпадающие с выведенными вашим решением при первом запуске.	
2	7	$f(x) = x + 2000$	$s \leqslant 200\,000$	$t_1 < t_2 < \ldots < t_n$	
3	6	$f(x) = x + 2000$	$s \leqslant 200\,000$	$n = 2$	
4	до 34	$f(x) = 4 \cdot x + 2000$	$s \leqslant 200\,000$		
5	до 11	$f(x) = x + 2000$		$t_1 = t_2 = \ldots = t_n > 1$	
6	до 9	$f(x) = x + 2000$		$t_i > 1$	5
7	до 20	$f(x) = x + 2000$			1 – 6

Четвертая подзадача оценивается по следующей формуле. Обозначим $k = max\left(0, \frac{m - 2000}{s}\right)$ для каждого набора входных данных. Введем функцию score(k) следующим образом:

k	score(k)
$\leqslant 1{,}5$	34
2	20
3	10
4	5
> 4	0

Для промежуточных значений k функция вычисляется линейно между соседними строками таблицы и округляется до ближайшего целого числа.

Балл за тест равняется минимуму score(k) по всем наборам входных данных в тесте. Балл за подзадачу равняется минимуму из баллов по тестам этой подзадачи.

Подзадачи 5, 6 и 7 также оцениваются по формуле. Обозначим $c = \max(0, m - s)$ для каждого набора входных данных. Введем функцию score(c) следующим образом:

c	score(c), подз. 5	score(c), подз. 6	score(c), подз. 7
$\leqslant 30$	11	9	20
100	7	7	14
200	4	4	8
2000	2	2	4
> 2000	0	0	0

Для промежуточных значений c функция вычисляется линейно между соседними строками таблицы и округляется до ближайшего целого числа.

Балл за тест равняется минимуму score(c) по всем наборам входных данных в тесте. Балл за подзадачу равняется минимуму из баллов по тестам этой подзадачи.

Пример


стандартный поток ввода
стандартный поток вывода
первый запуск
1
3
()
(())
(()())
(((()(()())))
1
((())())
(((((((((())))))))))
0
второй запуск
2
(((((((((())))))))))
1
(()(()))
((((()())()))
3
()+(())+(()())
0