

## Задача А. LCA

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дано подвешенное дерево с корнем в вершине 0. Вам нужно ответить на  $m$  запросов вида «найти LCA двух вершин». LCA вершин  $u$  и  $v$  в подвешенном дереве – это наиболее удаленная от корня дерева вершина, лежащая на обоих путях от  $u$  и  $v$  до корня.

### Формат входных данных

В первой строке вводится натуральное число  $n$  – размер дерева ( $2 \leq n \leq 10^5$ ). В следующей строке записано  $n - 1$  целое число  $p_i$  – предок вершины  $i$  ( $0 \leq p_i < i$ ).

Затем дано число  $m$ . Далее заданы  $m$  ( $0 < m \leq 10^5$ ) запросов вида  $(u, v)$  – найти LCA двух вершин  $u$  и  $v$  ( $0 \leq u, v < n; u \neq v$ ).

### Формат выходных данных

На каждый из  $m$  запросов выведите по одному числу – LCA заданных вершин.

### Примеры

стандартный ввод	стандартный вывод
5 0 0 1 2 2 1 2 3 4	0 0
5 0 0 1 1 3 3 4 3 1 2 4	1 1 0

## Задача В. Самое дешевое ребро

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на  $M$  запросов вида «найти у двух вершин минимум среди стоимостей ребер пути между ними».

### Формат входных данных

В первой строке задано целое число  $n$  – число вершин в дереве ( $2 \leq n \leq 10^5$ ).

В следующих  $n - 1$  строках записаны два целых числа  $x$  и  $y$ . Число  $x$  на строке  $i$  означает, что  $x$  – предок вершины  $i$ ,  $y$  задает стоимость ребра ( $x < i; |y| \leq 10^6$ ).

Далее заданы  $m$  ( $0 \leq m \leq 10^5$ ) запросов вида  $(x, y)$  – найти минимум на пути из  $x$  в  $y$  ( $0 \leq x, y < n; x \neq y$ ).

### Формат выходных данных

Выведите ответы на запросы.

### Примеры

стандартный ввод	стандартный вывод
5	2
0 2	2
0 3	
1 5	
2 2	
2	
1 2	
3 4	
5	1
0 1	1
0 2	
1 3	
2 4	
2	
0 3	
2 1	

## Задача С. Длина максимального подпалиндрома

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Палиндромом называется строка, которая одинаково читается как слева направо, так и справа налево. Подпалиндромом данной строки называется последовательность символов из данной строки, не обязательно идущих подряд, являющаяся палиндромом. Например, HELOLEH является подпалиндромом строки HTEOLFEOLEH. Напишите программу, находящую в данной строке подпалиндром максимальной длины.

### Формат входных данных

Во входном файле находится строка длиной не более 100 символов, состоящая из заглавных букв латинского алфавита.

### Формат выходных данных

Выведите на первой строке выходного файла длину максимального подпалиндрома, а на второй строке сам максимальный подпалиндром. Если таких подпалиндромов несколько, то ваша программа должна вывести любой из них.

### Примеры

стандартный ввод	стандартный вывод
HTEOLFEOLEH	7 HELOLEH
ABCDEF	1 F

## Задача D. Распил брусьев

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Вам нужно распилить деревянный брус на несколько кусков в заданных местах. Распиличная компания берет  $k$  рублей за распил одного бруска длиной  $k$  метров на две части.

Понятно, что различные способы распила приводят к различной суммарной стоимости заказа. Например, рассмотрим брус длиной 10 метров, который нужно распилить на расстоянии 2, 4 и 7 м, считая от одного конца. Это можно сделать несколькими способами. Можно распилить сначала на отметке 2 м, потом 4 и, наконец, 7 м. Это приведет к стоимости  $10+8+6=24$ , потому что сначала длина бруса, который пилили, была 10 м, затем она стала 8 м, и, наконец, 6 м. А можно распилить иначе: сначала на отметке 4 м, затем 2, затем 7 м. Это приведет к стоимости  $10+4+6=20$ , что лучше.

Определите минимальную стоимость распила бруса на заданные части.

### Формат входных данных

Первая строка входных данных содержит целое число  $L$  ( $2 \leq L \leq 10^6$ ) – длину бруса и целое число  $N$  ( $1 \leq N \leq 100$ ) – количество распилов. Во второй строке записано  $N$  целых чисел  $C_i$  ( $0 < C_i < L$ ) в строго возрастающем порядке – места, в которых нужно сделать распилы.

### Формат выходных данных

Выведите одно натуральное число – минимальную стоимость распила.

### Пример

стандартный ввод	стандартный вывод
10 3 2 4 7	20

## Задача Е. Упаковка символов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Билл пытается компактно представить последовательности прописных символов от A до Z с помощью упаковки повторяющихся подпоследовательностей внутри них. Например, один из способов представить последовательность AAAAАААААВАВАВССД — это 10(A)2(ВА)B2(C)D. Он формально определяет сжатые последовательности символов и правила перевода их в несжатый вид следующим образом:

- Последовательность, содержащая один символ от A до Z, является упакованной. Распаковка этой последовательности даёт ту же последовательность из одного символа.
- Если S и Q — упакованные последовательности, то SQ — также упакованная последовательность. Если S распаковывается в S', а Q распаковывается в Q', то SQ распаковывается в S'Q'.
- Если S — упакованная последовательность, то X(S) — также упакованная последовательность, где X — десятичное представление целого числа, большего 1. Если S распаковывается в S', то X(S) распаковывается в S', повторённую X раз.

Следуя этим правилам, легко распаковать любую заданную упакованную последовательность. Однако Биллу более интересен обратный переход. Он хочет упаковать заданную последовательность так, чтобы результирующая сжатая последовательность содержала наименьшее возможное число символов.

### Формат входных данных

В первой строке находится последовательность символов от A до Z.

Ограничения: длина исходной последовательности от 1 до 100.

### Формат выходных данных

В единственной строке выводится упакованная последовательность наименьшей длины, которая распаковывается в заданную последовательность. Если таких последовательностей несколько, можно выводить любую.

### Примеры

стандартный ввод	стандартный вывод
AAAAАААААВАВАВССД	9(A)3(ВА)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

## Задача F. Монетки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2.2 секунд  
Ограничение по памяти: 256 мегабайт

64 мегабайта

В Волшебной стране используются монетки достоинством  $A_1, A_2, \dots, A_M$ . Волшебный человечек пришел в магазин и обнаружил, что у него есть ровно по две монетки каждого достоинства. Ему нужно заплатить сумму  $N$ . Напишите программу, определяющую, сможет ли он расплатиться без сдачи.

### Формат входных данных

Сначала вводится целое число  $N$  ( $1 \leq N \leq 10^9$ ), затем — целое число  $M$  ( $1 \leq M \leq 10$ ) и далее  $M$  попарно различных целых чисел  $A_1, A_2, \dots, A_M$  ( $1 \leq A_i \leq 10^9$ ).

### Формат выходных данных

Выведите сначала  $K$  — количество монет, которое придется отдать Волшебному человечку, если он сможет заплатить указанную сумму без сдачи. Далее выведите  $K$  чисел, задающих достоинства монет. Если решений несколько, выведите вариант, в котором Волшебный человек отдаст наименьшее возможное количество монет. Если таких вариантов несколько, выведите любой из них.

Если без сдачи не обойтись, то выведите одно число 0. Если же у Волшебного человечка не хватит денег, чтобы заплатить указанную сумму, выведите одно число  $-1$  (минус один).

### Примеры

стандартный ввод	стандартный вывод
5 2 1 2	3 1 2 2
7 2 1 2	-1
5 2 3 4	0