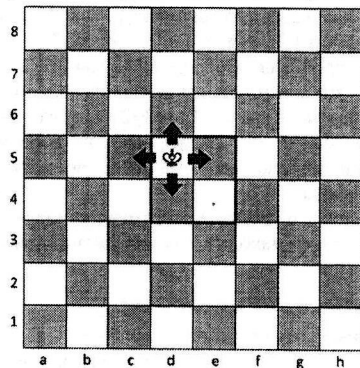


Задачи 7-8 класс

Задача 1. Шахматный король

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	1 мегабайт
Максимальная оценка:	100 баллов

Программист Юра очень любит шахматы. При этом самой любимой его фигурой является шахматный король, так как это самая главная фигура на доске, которая к тому же может ходить в любом из направлений (правда только на одну клетку).



Как-то раз, решая шахматные задачки, ему пришла в голову мысль: а сколько существует маршрутов для того, чтобы шахматный король попал из одной клетки доски в другую. При этом Юра решил ввести дополнительные ограничения:

- король при каждом своем ходе должен менять цвет клетки;
- король обязательно должен пройти через центральные клетки доски.

Маршрутом называется последовательность ходов ведущих из одной клетки шахматной доски в другую. Необходимо исключить «глупые» маршруты, т.е. нельзя делать ходы, которые увеличивают расстояние до конечной клетки маршрута. Под расстоянием понимается минимально возможное число ходов для достижения некоторой клетки. Каждая клетка на шахматной доске нумеруется с помощью двух координат: a, b, c, d, e, f, g, h – нумерация по горизонтали и цифры от 1 до 8 для нумерации по вертикали. Центральными клетками доски называют четыре клетки с координатами d4, d5, e4, e5.

Помогите Юре решить задачу. Требуется написать программу, которая будет определять количество маршрутов, по которым может пройти шахматный король, чтобы из одной клетки шахматной доски попасть в другую и при этом пройти хотя бы через одну центральную клетку. Если начальная и конечная клетки маршрута совпадают считать, что имеется один (пустой) маршрут тогда, когда эта клетка является одной из центральных.

Формат входных данных

Первая строка содержит натуральное число N ($1 \leq N \leq 1000$) - количество пар клеток на шахматной доске.

Следующие $N-1$ строка содержат пары клеток разделенные пробелом.

Формат выходных данных

В выходном файле для каждой пары клеток выводится количество маршрутов, которыми шахматный король может пройти от начальной клетки до конечной через центральные, либо слово NO, если таких маршрутов не существует.

Пример входных и выходных данных

input.txt	output.txt
2	NO
a1 c2	5
c4 e6	

input.txt	output.txt
3	1
d4 d4	65
h7 b4	NO
g2 h6	

Задача 2. Грубая сила

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	16 мегабайта
Максимальная оценка:	100 баллов

У Юры есть подруга Наташа, которой он часто помогает с домашним заданием по математике, физике, химии и информатике. В благодарность Наташа пригласила Юру вместе посмотреть с ней новый фильм. Да вот беда, Наташа забыла пароль для директории, в которой у нее хранятся фильмы. Но на всякий случай Наташа оставила себе подсказку – список символов, используемых в пароле. Наташа помнит, что каждый символ входил в парольное слово ровно 1 раз.

Юра хочет помочь подруге. Для этого он решил воспользоваться методом «грубой силы» (brute force) – метод поиска решения с перебором всех возможных вариантов. Однако, он забыл как реализовать этот метод и как узнать количество необходимых для проверки вариантов.

Требуется по заданному входному набору символов сформируйте все возможные комбинации кодовых слов, а также определите какое время потребуется на проверку всех значений паролей, если известно, что для проверки одного пароля требуется 1 с.

Формат входных данных

Во входном файле перечислены все символы пароля без повторов. В пароль могут входить буквы латинского алфавита в верхнем и нижнем регистрах, цифры от 0 до 9, а также запятая, тире, точка, восклицательный знак и символ подчеркивания. Число символов во входном наборе от 1 до 8.

Формат выходных данных

В первой строке выходного файла выводится количество секунд, которое нужно для проверки всех паролей (если секунд меньше 60), либо количество минут (округление вверх). В каждой следующей строке выводится возможный пароль. Пароли необходимо вывести в лексикографическом порядке (отсортированные по алфавиту в соответствии с кодами ASCII).

Пример входных и выходных данных

input.txt	output.txt
AB	2 AB BA
A1B	6 1AB 1BA A1B AB1 B1A BA1

Задача 3. Фибоначчиева система счисления

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта
Максимальная оценка:	100 баллов

У Юры есть друг Дима, который учится на первом курсе университета. Дима не любит программировать, но ему нравится математика, а один из любимых разделов Димы – теория чисел. Известной последовательностью чисел являются числа Фибоначчи – ряд чисел, получаемый по правилу:

$$f_i = f_{i-1} + f_{i-2}, f_0 = 1, f_1 = 1.$$

Фибоначчиева система счисления (ФСС) – это позиционная система счисления с алфавитом, состоящим из двух цифр: 0 и 1, а ее базисом является последовательность чисел Фибоначчи 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Дима пробовал переводить числа из десятичной системы счисления в ФСС и заметил, что это можно делать различными способами. Так, например, число 11_{10} может быть представлено и как $1111 (5+3+2+1)$, и как $10011 (8+2+1)$, и как $10100 (8+3)$. Диме это не понравилось и он решил ввести дополнительное ограничение для описания *канонической формы*: число в ФСС не должно содержать в любых двух соседних разрядах единицы.

Требуется написать программу, которая для всех десятичных чисел, переданных на вход программе, находит каноническую форму в ФСС или указывает, что такой формы нет.

Формат входных данных

В первой строке файла число $N (0 < N \leq 10^4)$ – количество чисел, которые будут переведены в ФСС. В следующих N строках находятся десятичные числа $A_1, A_2, \dots, A_N (0 \leq A_i \leq 10^5)$.

Формат выходных данных

В N строках выходного файла числа Фибоначчи (каноническая форма), либо слово NO, если соответствующее число не имеет канонической формы.

Пример входных и выходных данных

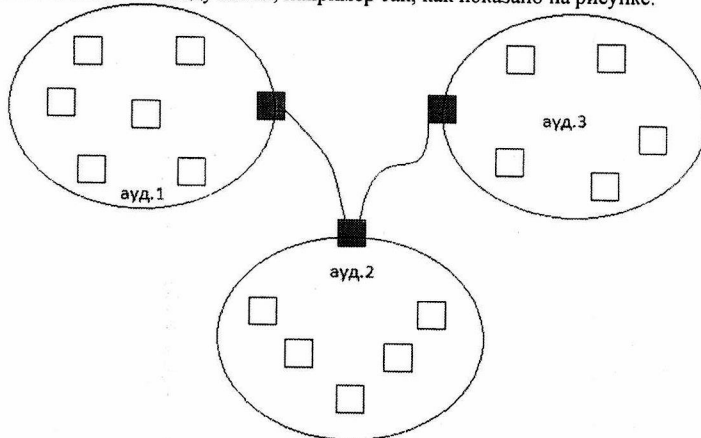
input.txt	output.txt
2	10100
11	100010
15	
3	1001000000
110	10000010001
153	100000
13	

Задача 4. Сеть университета

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта
Максимальная оценка:	100 баллов

В университете, в котором учится Дима, много компьютерных классов. В каждом из классов стоит специальное устройство - сетевой коммутатор (свитч). Свитч - это устройство, предназначенное для соединения нескольких узлов компьютерной сети, он хранит в своей памяти таблицу коммутации, что позволяет направлять трафик непосредственно компьютеру, которому он (трафик) предназначен, а не всем компьютерам (как это происходит в случае использования сетевого концентратора - хаба). Кроме того, современные свитчи могут также выполнять функции маршрутизации.

Все свитчи связаны между собой, например так, как показано на рисунке:



Как видно из рисунка, три аудитории соединяются тремя свитчами и двумя каналами (линиями). Каждый канал (линия) обеспечивает прямую пересылку данных между аудиториями в обе стороны. При этом изначально сетевая инфраструктура была спроектирована таким образом, чтобы из любого кабинета можно было послать пакет в любой кабинет. Назовем эту характеристику сети *связностью*.

В связи с увеличением производительности WiFi-оборудования и его широким распространением было решено прекратить использование проводных соединений после ремонта учебного корпуса, а имеющееся сетевое оборудование демонтировать. На демонтаж сетевое оборудование специалистам отводится ровно 1 день. Разумеется, что при демонтаже узлов, некоторые из еще работающих узлов могут стать недостижимыми, а сеть перестанет быть *связной* (например, на представленном рисунке при демонтаже первого узла 2 сеть перестанет быть связной).

Требуется по введенной информации об информационной сети разработать какой-либо порядок демонтаж сетевое оборудование по кабинетам, при котором сеть всегда будет оставаться связной, а значит, как можно большее количество кабинетов будут подключены в сеть как можно дольше.

Формат входных данных

Первая строка содержит два числа разделенных пробелами N ($0 \leq N \leq 10^3$) - число свитчей (кабинетов) и M ($0 \leq M \leq 10^6$) - количество прямых каналов соединения этих узлов. В следующих M строках находится информация об этих каналах, представленная парами чисел - номера связываемых каналом кабинетов.

Формат выходных данных

Выходной файл должен состоять из N строк, каждая из которых должна содержать одно число - номер кабинета. Вывод кабинетов осуществляется в порядке их демонтажирования. Среди равнозначных вариантов выбирается кабинет с меньшим номером.

Пример входных и выходных данных

input.txt	output.txt
5 4	1
3 1	2
3 2	4
3 4	3
3 5	5

Задача 5. Игра в слова

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	16 мегабайт
Максимальная оценка:	100 баллов

У Юры есть младший брат Вова. Вова еще не ходит в школу, но уже учит английские буквы и слова. Чтобы Вове было интереснее учиться, Юра подарил ему набор из кубиков. На каждом кубике есть картинка и слово, одна из букв которого больше остальных (а также выделена цветом). Таким образом, из выделенных букв можно складывать слова, при этом в качестве подсказки используя те слова, которые уже есть на кубиках.

Каждый такой кубик содержит шесть граней, на каждой одна картинка, одно слово и одна выделенная буква. При этом каждое слово на кубике не может повториться во всем наборе, а вот выделенные буквы могут повториться как на одном кубике, так и на разных кубиках в одном наборе.

Вова попросил поиграть Юру вместе с ним, но Юре нужно было делать домашнее задание, поэтому он предложил Вове игру по следующим правилам: он пишет на карточках слова, а Вова должен их выложить. Однако, очень быстро Юра заметил, что не любое слово может быть выложено с использованием набора кубиков.

Требуется написать программу, которая определяет, возможно ли выложить слово с использованием известного набора кубиков.

Формат входных данных

В первой строке входного файла число N ($0 \leq N \leq 300$) – количество кубиков в наборе. Во второй строке записано задуманное Юрой слово, состоящее только из заглавных букв латинского алфавита, не длиннее 50 символов. Следующие N строк содержат по 6 слов, в которых все буквы, кроме одной заглавной строчные.

Формат выходных данных

Выводится YES – если возможно сложить слово или NO – в обратном случае. Если сложить слово нельзя, то во второй строке вывести максимальное количество букв, которое удалось сложить

Пример входных и выходных данных

input.txt	output.txt
2 DIGITAL hEater Voltage bRush brUsh dRive microwaVe intercEpt stOve laWn Desk heAter Video	NO 1

Задачи 9-11 класс

Задача 1. Гипотеза Гольдбаха.

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	16 мегабайта
Максимальная оценка:	100 баллов

Юра, как и все программисты, очень хорошо знает математику и интересуется теорией чисел. Недавно Юра познакомился с гипотезой Гольдбаха, которая гласит, что любое четное число, которое больше двух, может быть представлено в виде суммы двух простых чисел, причем разложений может быть несколько.

Юра решил проверить гипотезу, для этого он подобрал пары простых чисел для нескольких небольших четных чисел: $4=2+2$, $6=3+3$, $8=3+5$, $10=3+7=5+5$, $12=5+7$, $14=7+7=3+11$ и т.д.

Требуется написать программу, которая подтверждает гипотезу Гольдбаха для переданных на вход чисел

Формат входных данных

Входной файл содержит в первой строке число N ($1 \leq N \leq 1000$), а затем еще N строк, в которых находятся числа M_i ($4 \leq M_i \leq 99998$).

Формат выходных данных

В выходном файле требуется вывести в каждой из N строк два простых числа, сумма которых равна числу M_i . Если пар несколько, то выводится та пара, в которой содержится наименьшее простое число. Первым в паре выводится наименьшее число.

Пример входных и выходных данных

input.txt	output.txt
4	107 99581
99688	3 1021
1024	2 11
13	2 233
235	
2	3 41
44	3 29
32	

Задача 2. Стройка тысячелетия.

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта
Максимальная оценка:	100 баллов

Вавилонская башня – это легендарное библейское сооружение. Согласно преданию в начале времен (после всемирного потопа) все люди разговаривали на одном языке. Род людской процветал и было решено построить город Вавилон с башней высотой до небес.

Но Богу не понравилась идея людей, ибо увидел он в ней гордыню. Бог решил помешать людям, но так, чтобы доказать это не представлялось возможным. Тогда Бог создал новые языки для разных людей.

В результате действий Бога оказалось, что каждый человек знает некоторое множество языков, которое является подмножеством всех языков придуманных Богом. Два человека могут передавать друг другу информацию только, если существует язык, который они оба знают.

Руководитель стройки передает команды на известных ему языках. Те, кто эти команды получил, могут их передавать дальше, переводя на известные им языки. Требуется написать программу, которая поможет определить будет ли стройка доведена до конца. Считается, что стройка будет доведена до конца, если команда руководителя может быть интерпретирована хотя бы 80% участников стройки.

Требуется написать программу, которая определит может ли стройка быть закончена вовремя и сколько участникам стройки доходят команды руководителя.

Формат входных данных

Для удобства пронумеруем все языки числами от 1 до 50. Во входном файле задано количество людей N ($1 \leq N \leq 100$), а дальше идут описания того, какие языки знают эти люди. Для каждого человека записано сначала число M_i , определяющее количество языков, известных i -ому человеку, а затем перечисляются номера самих этих языков в возрастающем порядке (номера языков — числа от 1 до 50). Считается, что руководитель строительства — это человек с номером 1.

Формат выходных данных

Выходной файл состоит из двух строк. В первой содержится одно слово: YES - стройка будет доведена до конца или NO - стройка не будет доведена до конца. Во второй выводится целое число – количество человек, до которых может «дойти» отданная руководителем команда (включая самого руководителя).

Пример входных и выходных данных

input.txt	output.txt
5	NO
2 1 2	3
1 1	
2 2 3	
0	
2 4 5	

8	YES
3 1 4 8	7
3 2 4 15	
3 12 14 19	
2 15 33	
2 8 11	
4 2 4 18 21	
1 15	
2 21 23	

Задача 3. Карточный фокус

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	1 мегабайт
Максимальная оценка:	100 баллов

Поручик Ржевский изобрел карточный фокус для удивления дам своей интуицией и феноменальной памятью. Для этого он берет пять карточных колод, выбрасывает из них все шестерки, а затем раскладывает карты по пяти колодам специальным образом:

Колода 1 ♠7 ♠7 ♠8 ♠8 ♠9 ♠9 ♠10 ♠10 ♠В ♠В ♠Д ♠Д ♠К ♠К ♠Т ♠Т
 Колода 2 ♠7 ♠7 ♥7 ♠7 ♠8 ♠8 ♥8 ♠8 ♠9 ♠9 ♥9 ♠9 ♠10 ♠10 ♥10 ♠10
 Колода 3 ♠7 ♥7 ♠8 ♥8 ♠9 ♥9 ♠10 ♥10 ♠В ♥В ♠Д ♥Д ♠К ♥К ♠Т ♥Т
 Колода 4 ♠7 ♠7 ♥7 ♠7 ♠9 ♠9 ♥9 ♠9 ♠В ♠В ♥В ♠В ♠К ♠К ♥К ♠К
 Колода 5 ♠7 ♠7 ♥7 ♠7 ♠8 ♠8 ♥8 ♠8 ♠В ♠В ♥В ♠В ♠Д ♠Д ♥Д ♠Д

Всего в карточной колоде используется 4 масти:

- ♠ - пики;
- ♣ - трефы;
- ♥ - черви;
- ♦ - бубна.

Для каждой масти 8 карт:

- 7 – семерка;
- 8 – восьмерка;
- 9 – девятка;
- 10 – десятка;
- В – валет;
- Д – дама;
- К – король;
- Т – туз.

Затем поручик предлагает даме задумать любую карту от семерки пик, до бубнового туза, а затем перечислить поручику те колоды, в которые входит задуманная карта. После этого поручик практически мгновенно выдает ответ и всегда угадывает.

Требуется написать программу, которая не сохраняя заданные 5ть колод по номерам колод, в которые входит загаданная карта определяет ее.

Формат входных данных

Во входном файле в одной строке перечисляются номера колод (от 1 до 5), в которых присутствует загаданная карта.

Формат выходных данных

Выходной файл должен содержать название задуманной карты.

Пример входных и выходных данных

input.txt	output.txt
3 5	Д черви
2 4 5	7 бубна

Задача 4. Контрольные коды

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	16 мегабайта
Максимальная оценка:	100 баллов

Алиса и Боб играют в шпионов и передают друг другу секретные сообщения. Для того, чтобы сообщение стало секретным каждую букву сообщения нужно перевести в специальную шестибитную кодировку (используется равномерный двоичный код и длина каждого символа должна быть минимальной), при этом используются только символы русского алфавита (нет различий по регистрам), пробел, и цифры от 0 до 9. Для передачи сообщения используется фонарик. Передача осуществляется по следующим правилам: в начале Алиса машет фонариком, что является сигналом о начале передачи. Затем начинается побитовая передача сообщения: если фонарик светит фронтально – 1, если фонарик отключен (прикрыт ладонью) – 0. На передачу каждого бита отводится ровно 1 секунда, при этом для отделения двух последовательно передаваемых бит фонарик направляется в сторону.

Для того, чтобы обезопасить себя от неверной интерпретации передаваемого сообщения, а также от его фальсификации третьими лицами Боб предложил Алисе использовать следующий признак: перед началом передачи в конец строки добавляется один контрольный символ, который определяет корректность передаваемого сообщения. Корректным считается сообщение, которое после перевода из двоичной системы в десятичную (в виде большого целого числа) делится без остатка на 31. Боб очень хорошо считает и быстро придумывает коды, а вот у Алисы есть определенные проблемы с арифметикой.

Требуется написать программу, которая будет проверять корректность передаваемых сообщений.

Формат входных данных

В первой строке входного файла число N – количество полученных сообщений. В следующих N строках приводятся сообщения, которые необходимо проверить в виде последовательности 0 и 1. Длина каждой такой последовательности не превышает 6000 символов (т.е. не более 99 символов в передаваемом сообщении).

Формат выходных данных

В выходном файле в N строках печатается одно из двух слов: YES – если сообщение успешно прошло проверку на корректность и NO – в обратном случае.

Пример входных и выходных данных

input.txt	output.txt
2 110111000110110110100100 110111000110110110110000	NO YES
3 110111000110110110100100 110111000110110110110000 010101010100000	NO YES YES

Задача 5. Минимальная кодировка

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	16 мегабайта
Максимальная оценка:	100 баллов

Энтропийное кодирование - это метод кодирования данных, который обеспечивает компрессию данных за счет удаления избыточной информации. Например, русский текст, закодированный с помощью таблицы ASCII, - это пример сообщения с высокой энтропией. В тоже время сжатые сообщения, например, zip-архивы, имеют очень маленькую энтропию, и потому попытки их энтропийного кодирования не принесут пользы.

Русский текст, закодированный с помощью ASCII, имеет высокую степень энтропии, потому что для кодирования всех символов используется одно и тоже количество бит - восемь. В то же время известный факт состоит в том, что буквы А, Е, И, О, Н, Т - встречаются чаще со значительно более высокой частотой, чем другие буквы русского алфавита. Если мы найдем способ закодировать только эти буквы четырьмя битами, то закодированный текст станет существенно меньше и при этом будет содержать всю исходную информацию и иметь меньшую энтропию. В результате кодирования часть букв останутся длинны из 8 бит, а часть - станут длиной 4 бита. Для того, чтобы можно было однозначно декодировать сообщения в данной кодировке будем использовать префиксное кодирование и условие Фано.

Условие Фано: *Никакое кодовое слово не может быть началом другого кодового слова.*

В такой схеме кодирования любое количество битов может быть использовано для конкретного символа. Однако для того, чтобы иметь возможность восстановить информацию, запрещено, чтобы последовательность битов, кодирующая некоторый символ, была префиксом битовой последовательности, используемой для кодирования любого другого символа. Таким образом, каждый символ читается бит за битом, пока не будет декодирован.

Например, у нас имеется сообщение:

никто не забыт, ничто не забыто

Данное сообщение содержит буквы: *н, и, к, т, о, е, з, а, б, ы, ч, ,* а также *запятую* и символ *пробела*. Всего получается - 13 символов (длина сообщения 31 символ). Для кодирования этого сообщения равномерным кодом нам бы потребовалось на каждый символ выделять не менее 4 бит:

н	и	к	т	о	е	з	а	б	ы	ч	,	
1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000

Тогда исходное сообщение может быть представлено последовательностью бит длиной: $4 * 31 = 124$ бита

110111001011101010010000110110000000011101100101010010100001000011011100
00111010100100001101100000000111011001010100101001

В восьмибитной кодировке данное сообщение занимает 248 бит.

Применим префиксную кодировку:

н	и	к	т	о	е	з	а	б	ы	ч	,	
011	0100	00010	110	001	0101	1010	1011	1000	1001	0000	00011	111

и получим сообщение (110 бит):

011010000010110001111011010111110101011100010011100001111101101000000110
00111101101011111010101110001001110001

Требуется написать программу, которая для входного текстового файла получает минимальный возможный размер в битах закодированного файла в случае использования оптимальной префиксной кодировки. Поскольку вариантов префиксной кодировки несколько то, для проверки оптимальности вашего варианта требуется вывести минимальное количество бит, которое потребуется для представления исходного сообщения в префиксной кодировке.

Формат входных данных

Содержит текст на любом языке, содержащий буквы русского и латинского алфавитов, знаки пунктуации, пробельные символы. Используется кодировка кодовая страница 1251.

Формат выходных данных

Минимальное число бит, требуемое на кодирование данного текста в соответствии алгоритмом энтропийного кодирования.

Пример входных и выходных данных

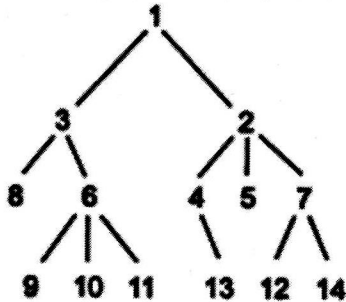
input.txt	output.txt
AAABCCD	13
никто не забыт, ничто не забыто	110

Задача 6. Спектр

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	16 мегабайта
Максимальная оценка:	100 баллов

Джеймс Бонд пытается разоблачить крупную преступную организацию с мировым влиянием, для чего ему приходится сотрудничать с ЦРУ. Преступная организация сильно законспирирована и состоит из рядовых членов и руководителей различных уровней. Во главе организации стоит один главный руководитель – лидер преступного мира. До начала арестов приказ лидера может быть доведен до любого члена организации. Все члены организации пронумерованы от 1 до N.

Каждый преступник знает только своего непосредственного босса (ровно одного) и своих непосредственных подчиненных (если они есть). В связи с мировым кризисом все страны участники борьбы преступностью стараются экономить деньги, поэтому Джеймсу Бонду не выдали оружия – он должен справляться с преступниками голыми руками.



Пример структуры преступной организации

Естественно, что с началом арестов вся организация начинает распадаться на мелкие, не связанные друг с другом группировки, которые функционируют в отдельных странах. Например, при аресте преступника №2 организация распадается на 4 группы. Бонд может победить голыми руками любую группу, состоящую из менее чем K преступников.

Для того, чтобы как можно быстрее и дешевле разоблачить и уничтожить всю преступную организацию перед сотрудниками ЦРУ была поставлена задача произвести минимальное количество арестов членов преступной организации таким образом, чтобы от нее остались только небольшие группы, с которыми Джеймс Бонд справится голыми руками.

Требуется написать программу, которая по входным данным описывающим преступную организацию определит количество необходимых арестов, а также тех, кого необходимо арестовать.

Формат входных данных

Входной файл содержит три строки. В первой записано число K ($1 \leq K \leq 10^5$) – определяющее максимальное число преступников в группе, при котором она может быть

ликвидирована Джеймсом Бондом. Во второй строке записано число N ($1 \leq N \leq 10^5$) – количество преступников. Третья строка содержит набор из N-1 числа. В этой строке для каждого члена преступной организации, кроме лидера, задается номер его непосредственного руководителя. Номер руководителя всегда меньше, чем номер подчиненного. Первое число задает номер руководителя второго преступника, второе – третьего и так далее. Числа в строке разделяются одним пробелом.

Формат выходных данных

В единственной строке выходного файла нужно напечатать минимальное число преступников, которое требуется арестовать.

Пример входных и выходных данных

input.txt	output.txt
4	4
20	
1 2 3 4 4 4 4 8 7 6 5 12 12 11 10 9 17 17 2	