

---

## Как осуществляется проверка

Входные данные подаются программе в стандартном потоке ввода (с клавиатуры). Программа должна выводить ответ в стандартный поток вывода (на экран). Если явно не оговорено в условии задачи, формат ввода предполагается свободным, то есть элементы входных данных могут отделяться друг от друга произвольным количеством пробельных символов. За все время тестирования разрешается отправить не более 200 решений суммарным размером не более 2 Мб. Размер исходного текста одного решения не должен превышать 64Кб.

Решение, отправленное на проверку, компилируется и прогоняется на заранее подготовленных тестах. Если решение на одном из тестов выдает неправильный ответ, тестирование прекращается. Вы можете посмотреть протокол тестирования и ознакомиться с тестом, на котором ваша программа работает не верно. Решение засчитывается только в том случае, если оно проходит все тесты.

При подведении результатов будут учитываться следующие параметры:

- количество решенных задач;
- качество программного кода;
- количество штрафных попыток.

## Задача А. Сканер

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

На белом фоне черным цветом написано число от 0 до 999. Каждый пиксел полученного изображения имеет белый, светло-серый, темно-серый или черный цвет. Ваша задача — определить число, которое изображено на картинке.

При генерации изображения используются стандартные системные шрифты размера от 20 до 50 пикселей.

## Формат входных данных

Картинка задается в виде ASCII графики. Белый цвет обозначен символом «.», светло-серый — «%», темно-серый — «#», черный — «@».

## Формат выходных данных

В выходной файл выведите искомое число.

## Примеры

[illegible]

### Примечание

В директории `samples` содержатся 18 тестов для отладки программы.

---

## Задача В. Расстояние в дереве

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан неориентированный связный граф без циклов (дерево), на ребрах которого написаны целые числа — расстояние между вершинами. Ваша задача — написать программу, которая умеет для заданной пары вершин определять длину кратчайшего пути.

### Формат входных данных

В первой строке входного файла записано целое число  $N$  — количество вершин в дереве. В последующих  $N - 1$  строках записано по три целых числа — информация о ребрах: начало, конец и длина ребра. В  $N + 1$  строке записано целое число  $M$  — количество пар вершин для которых нужно найти длину кратчайшего пути. В последующих  $M$  строках записаны пары целых чисел, определяющие номера вершин, между которыми необходимо вычислить расстояние.

Все числа во входном файле являются неотрицательными целыми, не превосходящие 40000.

### Формат выходных данных

В выходной файл выведите  $M$  целых чисел — длины кратчайших путей.

### Пример

входные данные	выходные данные
6	17
1 2 7	10
2 3 3	13
2 4 6	9
4 5 3	
5 6 1	
4	
1 6	
1 3	
6 3	
2 5	

#### Примечание

Эта задача является прямым следствием задачи о нахождении наименьшего общего предка в дереве (LCA). Подробную информацию по этому вопросу можно найти на сайте <http://e-maxx.ru>

---

## Задача С. Т9<sup>1</sup>

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Многие пользователи мобильных телефонов при наборе sms-сообщений используют режим Т9. При этом сообщения, например, на английском языке, они набирают следующим образом.

Для набора слова по буквам соответствующая букве кнопка нажимается один раз, вне зависимости от того, сколько букв соответствуют этой кнопке, и какой по счету идет нужная буква (см. картинку), а программа в телефоне подбирает из имеющегося словаря подходящее для данной комбинации кнопок слово. Если подходящих слов несколько, то в первую очередь предлагается наиболее часто встречающееся слово (изначально слова одинаковой встречаемости предлагаются по алфавиту).

1 .,?	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz

Если слово не подошло, то пользователь нажимает кнопку «\*» и программа предлагает второе по встречаемости слово, образуемое той же комбинацией кнопок (в первую очередь следующее слово с той же частотой встречаемости, если такое имеется). Если и оно не подошло, то кнопка «\*» нажимается еще раз и т.д. Для простоты будем считать, что современные модели телефонов содержат полный словарь используемых слов, и нужное слово обязательно найдется. Когда предлагаемое слово подошло, пользователь нажимает на кнопку «пробел», нажимает на кнопку «1» (последняя соответствует набору знака препинания), или заканчивает набирать сообщение. Когда знак препинания не подошёл, опять же нажимается кнопка «\*», до тех пор пока не появится требуемый знак. После набора пробела или знака препинания пользователь может ввести ещё один пробел или знак препинания, начать набирать следующее слово или закончить набирать сообщение. Будем считать, что пользователю достаточно трех знаков препинания, и они предлагаются в следующем порядке: точка «.», запятая «,», вопросительный знак «?».

После того как пользователь «утвердил» набранное слово (нажав на пробел или «1»), частота его встречаемости в словаре увеличивается на 1, и новое значение частоты учитывается, в том числе, и при наборе в режиме Т9 остальных слов того же сообщения. При этом данное слово будет предлагаться первым среди слов с такой же частотой встречаемости, порядок предложения остальных слов остается неизменным. Когда появится еще одно слово с той же частотой, то уже оно будет предлагаться первым, не меняя порядка остальных, и т. д.

Вам требуется написать программу, которая по имеющемуся словарю, содержащему первоначальные характеристики частоты встречаемости того или иного английского слова, и известной последовательности нажатий кнопок пользователем при наборе sms-сообщения в режиме Т9 произведет появившееся на экране сообщение.

### Формат входных данных

В первой строке входного файла находится целое число  $N$  ( $3 \leq N \leq 50000$ ) — количество слов в словаре. В каждой из следующих  $N$  строк записаны одно слово словаря и через пробел натуральное число  $F$  ( $1 \leq F \leq 1000$ ) — первоначальное значение частоты встречаемости этого слова (чем больше значение, тем чаще встречается данное слово). Числовая характеристика частоты встречаемости отделена от слова ровно одним пробелом. Слова в словаре состоят только из строчных английских букв и расположены в алфавитном порядке. Длина слова не превышает 20 символов. Все слова не пустые и различные.

---

<sup>1</sup>Задача взята с Международной Жаутыковской олимпиады  
<http://www.izho.kz/2010/problems.html>

---

Последняя строка файла состоит из цифр от 1 до 9 и символов «пробел» и «\*», обозначающих последовательность нажатий кнопок при наборе сообщения. Длина этой строки не превосходит 100000 символов.

## Формат выходных данных

Выведите в выходной файл текст sms-сообщения.

## Пример

входные данные	выходные данные
5 ad 2 be 1 not 10 or 5 to 50 86 23* 67 668 86 231**	to be or not to be?
3 act 1 bat 1 cat 1 228* 228** 228** 228**1	bat cat act bat.

### Пояснения к примерам

Во втором примере сначала программой будет предложено слово «**act**», затем «**bat**», которое будет принято пользователем. После чего частота встречаемости слова «**bat**» станет равной 2. Поэтому при повторном запросе той же числовой комбинации сначала предложено слово «**bat**», затем «**act**» и «**cat**». В третий раз слова будут предлагаться в порядке «**cat**» (как новое слово с частотой 2), «**bat**» (частота 2), «**act**» (и его частота сравняется с частотой остальных слов), и в последнем случае слова будут предлагаться в порядке «**act**», «**cat**», «**bat**».