

## Задача А. Код Хаффмана

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      512 мегабайт

Заданы числа  $p_1, p_2, \dots, p_n$ .

Предположив, что имеется текст, содержащий  $p_1$  символов  $c_1$ ,  $p_2$  символов  $c_2$ , и т. д., постройте код Хаффмана и найдите суммарное число битов, необходимое для кодирования такого текста.

### Формат входных данных

Первая строка ввода содержит число  $n$  ( $2 \leq n \leq 1000$ ). Вторая строка содержит  $n$  целых чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно число — число битов, необходимое для кодирования текста с заданным во входном файле количеством вхождений каждого символа.

### Пример

стандартный ввод	стандартный вывод
10 1 2 3 4 5 6 7 8 9 10	173

## Задача В. Преобразование Барроуза-Уиллера

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      512 мегабайт

Реализуйте преобразование Барроуза-Уиллера.

Рассмотрим строку  $s$ , состоящую из строчных латинских букв.

Отсортируем в лексикографическом порядке все ее циклические сдвиги. Выпишем последние буквы получившихся строк в порядке сортировки. Полученная строка называется преобразованием Барроуза-Уиллера заданной строки.

### Формат входных данных

Ввод содержит строку, содержащую не более 1000 строчных букв латинского алфавита.

### Формат выходных данных

Выведите результат преобразования Барроуза-Уиллера.

### Пример

стандартный ввод	стандартный вывод
abacaba	bcabaaa

## Задача С. Обратное преобразование Барроуза-Уиллера

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Реализуйте обратное преобразование Барроуза-Уиллера.

Рассмотрим строку  $s$ , состоящую из строчных латинских букв.

Отсортируем в лексикографическом порядке все ее циклические сдвиги. Выпишем последние буквы получившихся строк в порядке сортировки. Полученная строка называется преобразованием Барроуза-Уиллера заданной строки.

Вам дано преобразование Барроуза-Уиллера некоторой строки. Найдите эту строку. Поскольку у строк, одна из которых является циклическим сдвигом другой, преобразование Барроуза-Уиллера совпадает, из всех возможных ответов выведите лексикографически минимальный.

### Формат входных данных

Ввод содержит строку, содержащую не более 1000 строчных букв латинского алфавита. Гарантируется, что эта строка является преобразованием Барроуза-Уиллера некоторой строки.

### Формат выходных данных

Выведите минимальную лексикографически строку, для которой заданная является результатом преобразования Барроуза-Уиллера.

### Пример

стандартный ввод	стандартный вывод
bcabaaa	aabacab

## Задача D. Move To Front

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Реализуйте преобразование MTF.

Рассмотрим строку из строчных латинских букв.

Исходно буквы от 'a' до 'z' организованы в список в алфавитном порядке. По очереди рассматриваются буквы некоторого слова из латинских букв. Для каждой буквы кодируемой строки выполняется следующее:

- Выводится ее номер в списке (нумерация с 1).
- Она перемещается на первую позицию в списке.

### Формат входных данных

Входной файл содержит строку, содержащую не более 1000 строчных букв латинского алфавита.

### Формат выходных данных

Пусть длина строки во входном файле равна  $n$ . Выведите  $n$  чисел от 1 до 26, которые будут выведены при преобразовании Move To Front.

### Пример

стандартный ввод	стандартный вывод
abacaba	1 2 2 3 2 3 2

## Задача Е. Алгоритм LZW

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      512 мегабайт

Реализуйте кодирование в алгоритме LZW.

Рассмотрим строку  $s$ , состоящую из строчных латинских букв.

Исходно имеется словарь, содержащий символы от 'a' до 'z' с кодами от 0 до 25, соответственно.

Алгоритм поддерживает текущий буфер  $t$ , исходно инициализированный пустой строкой. Последовательно рассматриваются символы строки  $s$ . Пусть очередной символ строки равен  $c$ .

Если строка  $tc$  есть в словаре, то  $t$  присваивается  $tc$  и обработка символа завершается.

Иначе выводится код  $t$  и строка  $tc$  помещается в словарь с минимальным свободным кодом. После этого  $t$  присваивается значение  $c$  и обработка символа завершается.

После просмотра всех символов код оставшегося  $t$  также выводится.

### Формат входных данных

Входной файл содержит строку, содержащую не более 1000 строчных букв латинского алфавита.

### Формат выходных данных

Выведите коды, которые выводятся по мере выполнения алгоритма.

### Пример

стандартный ввод	стандартный вывод
abacaba	0 1 0 2 26 0

## Задача F. Раскодирование LZW

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Реализуйте декодирование в алгоритме LZW.

Напомним алгоритм LZW.

Рассмотрим строку  $s$ , состоящую из строчных латинских букв.

Исходно имеется словарь, содержащий символы от 'a' до 'z' с кодами от 0 до 25, соответственно.

Алгоритм поддерживает текущий буфер  $t$ , исходно инициализированный пустой строкой. Последовательно рассматриваются символы строки  $s$ . Пусть очередной символ строки равен  $c$ .

Если строка  $tc$  есть в словаре, то  $t$  присваивается  $tc$  и обработка символа завершается.

Иначе выводится код  $t$  и строка  $tc$  помещается в словарь с минимальным свободным кодом. После этого  $t$  присваивается значение  $c$  и обработка символа завершается.

После просмотра всех символов код оставшегося  $t$  также выводится.

### Формат входных данных

Первая строка ввода содержит число  $n$  — количество кодов LZW-кодировании ( $1 \leq n \leq 1000$ ). Вторая строка содержит  $n$  чисел — вывод алгоритма LZW. Гарантируется, что ввод является корректным LZW-кодом некоторой строки длины не больше 1000.

### Формат выходных данных

Выведите раскодированную строку.

### Пример

стандартный ввод	стандартный вывод
6 0 1 0 2 26 0	abacaba

## Задача G. Арифметическое кодирование

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

Дана строка над алфавитом, состоящим из первых  $n$  букв английского алфавита. Закодируйте ее с помощью арифметического кодирования.

При арифметическом кодировании отрезки при разбиении идут в порядке их следования в алфавите: сначала отрезок для буквы **a**, за ним отрезок для буквы **b**, и так далее. При нахождении дроби  $\frac{p}{2^q}$ , выбирается дробь с минимальным  $q$ , принадлежащая полуинтервалу  $[l, r)$ , полученному после обработки строки.

### Формат входных данных

В первой строке содержится число  $n$  — размер алфавита ( $1 \leq n \leq 26$ ). Во второй строке находится строка  $S$ , состоящая из первых  $n$  букв английского алфавита ( $1 \leq |S| \leq 100$ ). Некоторые из букв могут не присутствовать в  $S$ .

### Формат выходных данных

В первой строке выведите число  $n$ . Во второй строке выведите  $n$  чисел  $c_a, c_b, \dots$  — количество раз, которое в строке встречается каждая из  $n$  букв. В третьей строке выведите двоичное представление числителя  $p$  дроби  $\frac{p}{2^q}$ , ровно из  $q$  ( $q \geq 1$ ) нулей и единиц, при необходимости, с ведущими нулями.

### Примеры

стандартный ввод	стандартный вывод
3 abacaba	3 4 2 1 0110100101
2 bbb	2 0 3 0

## Задача Н. Арифметическое декодирование

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

Дан результат применения арифметического кодирования к некоторой строке  $s$ , состоящей из маленьких букв английского алфавита. Декодируйте и восстановите исходную строку.

### Формат входных данных

В первой строке содержится число  $n$  ( $1 \leq n \leq 26$ ), обозначающее, что в исходной строке присутствовали только первые  $n$  букв английского алфавита. Во второй строке содержатся  $n$  чисел  $c_a, c_b, \dots$  — количество раз, которое в строке встречается каждая из  $n$  букв ( $1 \leq \sum c_i \leq 100$ ). В третьей строке содержится строка из  $q$  ( $1 \leq q \leq 1000$ ) нулей и единиц — двоичное представление числителя  $p$  дроби  $\frac{p}{2^q}$ , возможно, с ведущими нулями.

### Формат выходных данных

Выведите одну строку длины  $\sum c_i$  — результат декодирования.

### Примеры

стандартный ввод	стандартный вывод
3 4 2 1 0110100101	abacaba
2 0 3 0	bbb



## Задача I. Исправление одной ошибки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Это задача с двойным запуском.

Ваше решение будет запущено два раза.

При первом запуске ему на вход будет передана строка  $x$  из нулей и единиц длиной не больше  $n$ . Программа должна вывести строку  $y$  из нулей и единиц, длина которой не более чем  $m$ . Число  $m$  не известно вашей программе и зависит от подзадачи, соответствующее значение указано в таблице системы оценивания. Если ваша программа выведет строку длиннее, чем  $m$ , она получит вердикт «Wrong answer».

Между запусками решения программа жюри внесет в строку  $y$  не более одной модификации, заменив ноль на единицу или единицу на ноль, получив, таким образом, строку  $z$ .

При втором запуске программе на вход будет подана строка  $z$ . Она должна восстановить исходную строку  $x$  и выдать её на выход.

### Формат входных данных

При первом запуске на первой строке ввода находится число 1. На второй строке ввода находится строка  $x$  из нулей и единиц длины  $n$  ( $10 \leq n \leq 100\,000$ ).

При втором запуске на первой строке ввода находится число 2. На второй строке ввода находится строка  $z$  из нулей и единиц длины не больше  $m$  ( $10 \leq m \leq 100\,017$ ). Гарантируется, что эта строка равна строке  $y$ , выведенной программой при первом запуске, или получена из неё изменением ровно одного символа на противоположный.

### Формат выходных данных

При первом запуске необходимо вывести строку  $y$ , которая позволит восстановить  $x$  после внесения в нее изменения. Длина строки  $y$  не должна превышать  $m$ .

При втором запуске по заданной строке  $z$  необходимо восстановить исходную строку  $x$  и вывести её.

## Примеры

Обратите внимание, что в примерах приведены конкретные варианты вывода и ввода при втором запуске, если ваша программа выведет другую строку  $y$ , при втором запуске ввод также может быть другим.

### Пример 1

Первый запуск.

стандартный ввод	стандартный вывод
1 0000000000	000000000000000000000000000000

Второй запуск.

стандартный ввод	стандартный вывод
2 000000000000000000000000000000	0000000000

### Пример 2

Первый запуск.

стандартный ввод	стандартный вывод
1 0000000000	000000000000000000000000000000

Второй запуск.

стандартный ввод	стандартный вывод
2 000000000000100000000000000000	0000000000