

## Задача А. Беги, Альф! Беги!

Имя входного файла: `run.in`  
Имя выходного файла: `run.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Прямо сейчас Альфу снится кошмар. В нем он бежит по дороге с препятствиями, на которой, ко всему прочему, разбросаны монеты.

Дорога представляет из себя таблицу  $n \times 3$ , в клетках которой либо ничего нет, либо находится стена, либо монета. Альф бежит вдоль стороны длиной  $n$ . Начинает он бежать из первой строки (то есть у него есть три варианта начала, он может выбрать любой из них) и бежит до тех пор, пока не врежется в стену, либо не пробежит дорогу целиком (не окажется в строчке  $n$ ).

Пусть сейчас Альф стоит в строке  $x$  и столбце  $y$  —  $(x; y)$ , тогда он может попасть в три возможные клетки:  $(x + 1; y - 1)$ ,  $(x + 1; y)$ ,  $(x + 1; y + 1)$ , если конечно новая клетка не выходит за пределы дороги, и в ней не находится стена. Так как все обитатели планеты Мелмак умеют контролировать свои сны, Альф смог получить карту дороги. Теперь он хочет узнать, какое наибольшее количество монет можно собрать к концу забега.

Так как контроль сна отнимает у Альфа много сил, он просит вас написать программу, которая по карте сможет определить наибольшее количество монет, которое можно собрать за один забег.

### Формат входного файла

В первой строке входного файла задано число  $n$  ( $1 \leq n \leq 10^4$ ) — количество строк в таблице. В следующих  $n$  строках дано по три символа, характеризующие данную строку таблицы.  $c$  равен «.», если клетка пустая, «С», если в этой клетке монета, и «W», если стена. Если в первой строке во всех клетках находятся стены, Альф заканчивает забег сразу.

### Формат выходного файла

В выходной файл выведите одно число — наибольшее количество монеток, которые можно собрать.

### Пример

<code>run.in</code>	<code>run.out</code>
5 W.W C.C WW. CC. CWW	3
4 W.W CWC W.W CWW	2

## Задача В. Граф

Имя входного файла: `graph.in`  
Имя выходного файла: `graph.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Много лет назад Альфу на день рождения подарили неориентированный граф. Недавно, разбирая вещи, он нашел его, и сильно удивился. Поиграв с ним некоторое время, ему стало интересно: сколько в этом графе есть пар вершин, между которыми нет ребра, а если его провести, в графе появится ровно один новый вершинно-простой цикл?

### Формат входного файла

В первой строке входного файла заданы два целых числа  $n, m$  ( $1 \leq n, m \leq 10^5$ ) — число вершин и ребер в графе соответственно.

В следующих  $m$  строках заданы пары целых чисел  $u, v$  ( $1 \leq u, v \leq n$ ) — ребра графа.

### Формат выходного файла

В единственной строке выходного файла выведите искомое число пар вершин.

### Пример

<code>graph.in</code>	<code>graph.out</code>
5 4 1 2 2 3 3 4 4 5	6
5 5 1 2 2 3 1 3 3 4 4 5	1

## Задача С. Маленькая шалость

Имя входного файла: `joke.in`  
Имя выходного файла: `joke.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Однажды ночью Альф решил пойти погулять. Во время прогулки пришельцу стало скучно, и он решил как-нибудь напакостить. А именно, увидев карту города, Альф решил перекрыть дорогу, причем не любую, а ту, после перекрытия которой кратчайшее расстояние от его дома до всех интересных мест города изменится у максимального числа интересных мест.

Город представлен в виде графа с  $n$  вершинами. Дом Альфа находится в вершине с номером 1.

Выведите максимальное число вершин, расстояние до которых изменится после удаления одного ребра в графе.

### Формат входного файла

В первой строке входного файла дано число  $n$  ( $1 \leq n \leq 300$ ) — количество вершин. В следующих  $n$  строках дано по  $n$  чисел  $a_{i,j}$  ( $-1 \leq a_{i,j} \leq 100000$ ) — матрица смежности. Если ребро в графе отсутствует  $a_{i,j} = -1$ . Гарантируется, что  $a_{i,i} = 0$ ,  $a_{i,j} > 0$  если  $i \neq j$ .

### Формат выходного файла

Выведите ответ на задачу.

### Пример

<code>joke.in</code>	<code>joke.out</code>
5 0 16 12 1 12 16 0 12 13 -1 12 12 0 5 2 1 13 5 0 2 12 -1 2 2 0	4

## Задача D. Рутинная работа

Имя входного файла: `stacks.in`  
Имя выходного файла: `stacks.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вчера Альф в очередной раз провинился — опять чуть не съел Лаки. На этот раз Вилли решил проучить Альфа.

В кладовке у него как раз завалились  $n + 1$  очередь и  $n$  стеков. Причем в одной из очередей также находилось  $2^n$  различных целых чисел от 1 до  $2^n$ . Вилли выложил перед Альфом все очереди и стеки в чередующемся порядке — сначала идет очередь, потом стек, потом опять очередь, и так далее, последней Вилли выложил очередь. Причем очередь с числами оказалась первой.

Все, что Альф может делать с этими стеками и очередями — вынуть из структуры данных номер  $i$  первое число (для стека это число на вершине, для очереди — число, находящееся в голове) и добавить его в структуру данных номер  $i + 1$  (в случае стека число добавится на его вершину, в случае очереди — в хвост). Теперь Вилли просит Альфа отсортировать все числа из первой очереди — сделать несколько операций над этими структурами данных, чтобы все числа находились в последней очереди, а также располагались там в возрастающем порядке. То есть должно быть выполнено, что в голове очереди находится число 1, в хвосте очереди —  $2^n$ , а между ними числа должны быть отсортированы.

Вилли утверждает, что отсортировать числа гарантированно можно за  $2 \cdot 2^n \cdot n$  операций. Помогите Альфу понять, как нужно применять эти операции, чтобы выполнить задание Вилли.

### Формат входного файла

В первой строке входного файла дано целое число  $n$  ( $1 \leq n \leq 15$ ) — число стеков. Во второй строке входного файла дано  $2^n$  различных чисел  $a_i$  ( $1 \leq a_i \leq 2^n$ ) — исходный набор чисел.

### Формат выходного файла

В первой и единственной строке выходного файла выведите  $2 \cdot 2^n \cdot n$  чисел  $b_i$  ( $1 \leq b_i \leq 2n$ ) — номер структуры данных, из которой извлекается число на  $i$ -м шаге вашего решения. Данная структура не должна быть пуста.

### Пример

<code>stacks.in</code>	<code>stacks.out</code>
1	1 2 1 2
1 2	
1	1 1 2 2
2 1	

### Комментарий

<https://ru.wikipedia.org/wiki/Стек>

### Комментарий

[https://ru.wikipedia.org/wiki/Очередь\\_\(программирование\)](https://ru.wikipedia.org/wiki/Очередь_(программирование))

## Задача E. Альф и доска с мелом

Имя входного файла: `blackboard.in`  
Имя выходного файла: `blackboard.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Альф впервые в жизни увидел доску для класса и осознал, что на ней можно писать мелом все, что душе угодно, по желанию стирать и писать заново.

Альф написал на доске число  $n$ . Далее он решил развлечься, поэтому каждую минуту он прибавляет к числу его последнюю цифру, стирает старое число и записывает новое. Альф довольно быстро устал, но так как он очень любопытный, он хочет узнать, какое число он должен был бы записать через  $k$  минут.

Альф просит вас помочь, иначе, если у вас есть кошка или кот, он придет и съест ее, ну или просто расстроится.

### Формат входного файла

В единственной строке входного файла даны два числа  $n$  и  $k$  ( $0 \leq n, k \leq 10^9$ ) — начальное число и количество минут соответственно.

### Формат выходного файла

В единственной строке выходного файла выведите одно число  $x$  — число, которое будет написано на доске, через  $k$  минут.

### Пример

<code>blackboard.in</code>	<code>blackboard.out</code>
1 10	44
5 1	10

## Задача F. Задача для Альфа

Имя входного файла: `maxnumber.in`  
Имя выходного файла: `maxnumber.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Чтобы хоть как-то занять Альфа, Вилли Таннер предложил ему любопытную задачу.

У Альфа есть  $n$  неотрицательных чисел. Каждое число можно приписать в конец другому и получить новое число. Например, если есть числа 123 и 456, из них можно получить либо 123456, либо 456123. Задача состоит в том, чтобы, объединив все числа в одно, таким образом получить наибольшее возможное. Так, если у Альфа есть два числа 123 и 456, то ответ на задачу будет 456123.

Вилли схитрил и дал Альфу очень много чисел, но телевизор сам себя не посмотрит, поэтому Альф просит вас написать программу, которая решит эту задачу.

### Формат входного файла

В первой строке входного файла задано число  $n$  ( $1 \leq n \leq 10^5$ ) — количество чисел данных Альфу. Во второй строке файла через пробел даны  $n$  чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — данные Альфу числа.

### Формат выходного файла

В выходной файл через пробел выведите  $n$  чисел из входного файла в таком порядке, чтобы при их последовательном соединении получалось наибольшее из возможных чисел.

### Пример

<code>maxnumber.in</code>	<code>maxnumber.out</code>
3 1 2 3	3 2 1
4 20 17 18 2	2 20 18 17

## Задача G. Ученье — свет, а неученье — тьма

Имя входного файла: `divisors.in`  
Имя выходного файла: `divisors.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 512 мегабайт

Последнее время Альф вообще ничего не делает. Даже ничего не сломал за последнюю неделю. Линн решила, что ему пора прекратить бездельничать и начать учиться.

Начать обучение она решила с несложной задачи: дан массив натуральных чисел. С массивом производятся две операции:

- `0 i x` — присвоить  $i$ -му элементу массива значение  $x$ ;
- `1 l r` — вывести количество делителей у произведения чисел на отрезке с  $l$  по  $r$ . Это число может быть довольно большим, поэтому ответ требуется вывести по модулю  $10^9 + 7$ .

Альф тайно влюблен в Линн, поэтому не может ей отказать. Но и работать он не хочет. Поэтому он попросил вас в тайне от Линн написать программу, которая решает данную задачу.

### Формат входного файла

В первой строке входного файла дано число  $n$  ( $1 \leq n \leq 5 \cdot 10^4$ ) — количество элементов массива.

Во второй строке дано  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 10^4$ ) — элементы массива.

В третьей строке дано число  $q$  ( $1 \leq q \leq 10^4$ ) — количество запросов.

В каждой из следующих  $q$  строк записано по три числа. Если первое из этих чисел равно 0, то это запрос обновления элемента массива, если же первое число равно 1, то это запрос на нахождение количества делителей у произведения чисел на отрезке по модулю  $10^9 + 7$ .

Гарантируется, что во всех запросах  $1 \leq i, l, r \leq n; 1 \leq x \leq 10^4$ .

### Формат выходного файла

На каждый запрос нахождения количества делителей выведите в отдельной строке выходного файла ответ — количество делителей у произведения всех чисел на данном отрезке по модулю  $10^9 + 7$ .

### Пример

<code>divisors.in</code>	<code>divisors.out</code>
5	12
2 3 4 5 6	6
6	6
1 2 4	12
1 2 3	
0 1 1	
0 4 7	
1 1 3	
1 1 4	

## Задача Н. Любимая строка

Имя входного файла: `string.in`  
Имя выходного файла: `string.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В очередной раз копаясь в гараже Вилли, Альф нашел строку, которую, судя по всему, хозяин гаража мастерил с особым усердием. Любознательный пришелец, как обычно, не растерялся и решил поиграть с этой строкой. Он разломал ее на  $k$  частей одинаковой длины и разбросал по всему гаражу. Вилли, увидев что случилось с его строкой, пришел в ярость. Свою злость он выместил на Альфе, приказав ему склеить строку обратно. К счастью для пришельца, у Вилли нашелся чертеж этой строки.

Помогите Альфу воссоздать строку по кускам и чертежу!

### Формат входного файла

В первой строке входного файла даны два числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^6$ ) — длина строки и количество кусков соответственно. Во второй строке входного файла дана строка  $s$  ( $1 \leq |s| \leq 10^6$ ) — строка, которую необходимо получить. В  $i$ -й из следующих  $m$  строк дана строка  $t_i$  — описание  $i$ -го куска исходной строки. Гарантируется, что  $n$  делится на  $m$  и из данных кусков можно составить исходную строку.

### Формат выходного файла

В единственной строке выходного файла выведите  $m$  различных целых чисел  $a_i$  ( $1 \leq a_i \leq m$ ), таких, что если заменить число на  $i$ -й позиции куском с номером  $a_i$  и склеить получившиеся куски, получится исходная строка.

Если ответов несколько, выведите любой.

### Пример

<code>string.in</code>	<code>string.out</code>
12 3 cabacaqwerty erty caba caqw	2 3 1