

## Задача A. Exotic Language

Имя входного файла:	<code>brainf.in</code> или <i>стандартный ввод</i>
Имя выходного файла:	<code>brainf.out</code> или <i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В новом, 2013 году группа пользователей популярного ресурса договорилась со спонсорами (в лице туристических компаний) о старте экзотического соревнования по спортивному программированию. После традиционных отборочных раундов планируется проведение онсайт-раунда на побережье одной из бывших британских колоний, с «Юнион Джеком» и шляпками, а в декабре — выдвижение соревнования на номинацию «Проект года».

Вы включены в оргкомитет соревнования. На данный момент ваша задача — написать интерпретатор с одного экзотического языка программирования.

В этом языке программирования память представлена в виде одномерного массива из  $3 \cdot 10^4$  беззнаковых байт ( $255 + 1 = 0$ ,  $0 - 1 = 255$ ), имеется указатель на текущую ячейку. Изначально во всех ячейках записаны нули, а указатель соответствует самой левой ячейке.

Программа представляет собой строку, каждый символ которой представляет собой одну из восьми команд:

- `'>'` — сдвигает указатель на одну ячейку вправо;
- `'<'` — сдвигает указатель на одну ячейку влево;
- `'+'` — прибавляет единицу к текущему значению ячейки;
- `'-'` — отнимает единицу от текущего значения ячейки;
- `'['` — исполнение переходит к символу, следующему за парной закрывающейся скобкой, если текущее значение ячейки равно нулю;
- `']'` — исполнение переходит к парной открывающейся скобке, если текущее значение ячейки не равно нулю;
- `'.'` — выводит символ с кодом, равным содержимому текущей ячейки;
- `'#'` — останавливает выполнение программы.

Остальные символы в программе игнорируются. Символы считываются слева направо по одному (за исключением случаев действия квадратных скобок).

### Формат входных данных

Входной файл содержит программу на вышеописанном языке программирования. Она может быть разорвана на несколько строк, индикатором завершения программы является знак решётки (`'#'`).

Гарантируется, что поданная на вход программа является корректной, будет работать разумное время и не выйдет за левую или правую границу массива. Размер входного файла не превышает  $10^4$  байт.

### Формат выходных данных

Ваша программа должна вывести то же, что и заданная во входном файле программа. Гарантируется, что все выводимые символы или будут переводами строк, или будут лежать в интервале от 32 до 127.

## Примеры

brainf.in или стандартный ввод
<pre>++[&gt;+++++&lt;-] // 26 to first cell &gt;&gt;&gt;+++++[&lt;++++&gt;-]&lt;+ : 65 to cell 2 &lt;[-&gt;.&lt;] /* doing a work */ +++++. final line break; and finish with #...</pre>
brainf.out или стандартный вывод
ABCDEFGHIJKLMNOPQRSTUVWXYZ

## Задача В. Communications

Имя входного файла: `communications.in` или **стандартный ввод**  
Имя выходного файла: `communications.out` или **стандартный вывод**  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Наилучшее место для проведения экзотических соревнований по программированию — какая-нибудь тропическая страна. Вот и в этот раз организаторы постарались и выбрали для проведения соревнований архипелаг, состоящий из нескольких атоллов. Каждый атолл представляет собой многоугольник без самопересечений. Никакие два атолла не имеют общей точки.

Между атоллами курсирует водное такси, водители которого взимают плату в соотношении один байтландский тугрик за одну байтландскую милю пути.

В данный момент участник находится на некотором атолле  $A$ . Открытие состоится в здании, расположенном на атолле  $B$ . За какую минимальную сумму участник сможет добраться на открытие, если за передвижение по суше плата не взимается?

### Формат входных данных

В первой строке входного файла задано целое число  $T$  — количество тестовых примеров ( $1 \leq T \leq 100$ ). В первой строке каждого тестового примера задано число  $n$  ( $2 \leq n \leq 50$ ) — количество атоллов. Во второй строке заданы два целых числа  $a$  и  $b$  ( $1 \leq a, b \leq n$ ,  $a \neq b$ ) — номера соответственно атолла, на котором расположен участник, и атолла, на котором пройдёт открытие.

Далее описаны атоллы в порядке возрастания их номеров (начиная с первого). В первой строке описания атолла содержится целое число  $v_i$  ( $3 \leq v_i \leq 50$ ) — количество вершин многоугольника, задающего атолл, далее в  $v_i$  строках заданы пары целых чисел  $(x_{ij}, y_{ij})$ , по модулю не превосходящие  $10^4$  — координаты  $j$ -й вершины атолла. Вершины перечислены в порядке обхода против часовой стрелки.

Гарантируется, что входной файл содержит не более  $32 \cdot 10^3$  строк.

### Формат выходных данных

Для каждого тестового примера в отдельной строке выведите минимальную сумму, которую должен заплатить участник, с точностью не хуже  $10^{-3}$ .

### Пример

<code>communications.in</code> или стандартный ввод	<code>communications.out</code> или стандартный вывод
1	5.657
3	
1 3	
4	
1 1	
3 1	
3 3	
1 3	
4	
5 5	
7 5	
7 7	
5 7	
4	
9 1	
11 1	
11 3	
9 3	

## Задача C. Race

Имя входного файла: `race.in` или *стандартный ввод*  
Имя выходного файла: `race.out` или *стандартный вывод*  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Участникам экзотического соревнования по программированию было предложено посетить традиционные для жителей архипелага гонки на волах. Всего на старте находятся  $n$  волов. Скорость  $i$ -го вола постоянна и равна  $v_i$  метров в минуту. Стартовые позиции волов определяются жеребьёвкой:  $i$ -й вол находится на расстоянии  $h_i$  метров от старта по направлению движения.

Основным недостатком такой гонки является то, что, начиная с некоторого момента, волы перестанут обгонять друг друга и гонка будет скучной. Поэтому среди аборигенов выбирают судью, останавливающего гонку сразу по завершении минуты, во время которой произошёл последний обгон (по традиции, гонка длится целое число минут).

Определите, после какой минуты судья остановит гонку.

### Формат входных данных

В первой строке входного файла дано целое число  $n$  — количество волов на старте ( $1 \leq n \leq 10^5$ ). В следующих  $n$  строках даны по два целых числа  $h_i$  и  $v_i$  — расстояние  $i$ -ого вола до старта и его скорость соответственно ( $0 \leq h_i, v_i \leq 10^9$ ).

### Формат выходных данных

В выходной файл выведите единственное число — номер минуты, после которой судья остановит гонку.

### Примеры

<code>race.in</code> или стандартный ввод	<code>race.out</code> или стандартный вывод
3 5 2 7 1 1 3	5
1 0 0	0
2 5 1 6 1	0

## Задача D. Seating

Имя входного файла:	<code>seating.in</code> или <i>стандартный ввод</i>
Имя выходного файла:	<code>seating.out</code> или <i>стандартный вывод</i>
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Опрос  $n$  участников экзотического соревнования по программированию показал, что участники будут использовать 26 различных языков программирования, при этом каждый участник будет использовать ровно один язык (который он знает лучше всего). Жюри соревнования обозначило эти языки строчными латинскими буквами от 'a' до 'z'. Рабочие места участников расположены по кругу. Два способа рассадки являются различными, если нельзя выбрать участника (для каждой рассадки — отдельно) и направление по часовой стрелке или против часовой стрелки (для каждой рассадки — своё) так, чтобы прочитанные в результате полного перечисления участников в заданном направлении в порядке рассадки строки, составленные из идентификаторов языков программирования, совпадали. В противном случае способы являются идентичными.

Вам заданы два способа рассадки. Требуется определить, являются ли они идентичными.

### Формат входных данных

В первой строке входного файла задано целое число  $T$  — количество тестовых примеров ( $1 \leq T \leq 100$ ). Каждый тестовый пример представляет собой две непустые строки длиной не более  $10^5$  каждая — идентификаторы участников, заданные в порядке рассадки, перечисленные, начиная с участника с идентификатором 0 по часовой стрелке, для каждой из сравниваемых рассадок.

### Формат выходных данных

Для каждого тестового примера выведите "YES", если заданные способы рассадки идентичны, и "NO" в противном случае

### Пример

<code>seating.in</code> или стандартный ввод	<code>seating.out</code> или стандартный вывод
5	YES
icpc	NO
pcic	YES
snws	YES
snss	NO
hijklmn	
lmnhijk	
ckbrainfu	
kcufniarb	
aaa	
aaaa	

## Задача E. The Results

Имя входного файла: `theresults.in` или *стандартный ввод*  
Имя выходного файла: `theresults.out` или *стандартный вывод*  
Ограничение по времени: 7 секунд  
Ограничение по памяти: 256 мегабайт

На одном экзотическом соревновании по спортивному программированию вручаются дипломы  $m + 1$  степеней, причём каждый участник получает диплом. Распределение дипломов происходит следующим образом. Пусть  $s_1, s_2, \dots, s_n$  — штрафные баллы, набранные каждым из  $n$  участников по результатам соревнования. Штрафные баллы являются целыми неотрицательными числами, не превосходящими  $10^4 - 1$ . Пусть  $L_0 = 0$  и  $L_{m+1} = 10^4$ ; тогда для  $i$  от 1 до  $m$  выберем  $L_i$  так, что если участник набрал  $L_i \leq s < L_{i+1}$  штрафных баллов, он получает диплом  $i$ -й степени.

Обозначим количество дипломов  $i$ -й степени за  $n_i$ , а долю дипломов  $i$ -й степени (равную  $n_i/n$ ) за  $p_i$ . Председатель жюри хочет выбрать  $L_i$  так, чтобы максимизировать энтропию распределения дипломов, равную  $-(p_0 \cdot \log_2(p_0) + p_1 \cdot \log_2(p_1) + \dots + p_m \cdot \log_2(p_m))$ . Считать, что в случае  $p_i = 0$   $0 \cdot \log_2(0) = 0$ .

Пока жюри выбирает значения  $L_i$ , Вам поручено написать программу, вычисляющую соответствующий максимум энтропии.

### Формат входных данных

Входной файл состоит из не более 30 тестовых примеров. Каждый тестовый пример начинается строкой, в которой заданы два целых числа  $n$  — количество участников и  $m$  — количество «отсечек» ( $1 \leq n \leq 1000$ ,  $0 \leq m \leq \min(n, 100)$ ). В следующей строке тестового примера заданы  $n$  целых неотрицательных чисел, меньших  $10^4$  — набранные участниками штрафные баллы. Входной файл завершается тестовым примером с  $m = n = 0$ , обрабатывать который не требуется.

Гарантируется, что входной файл содержит не более  $11 \cdot 10^3$  чисел.

### Формат выходных данных

Для каждого тестового примера выведите в отдельной строке максимальное значение энтропии с точностью не хуже  $10^{-2}$ .

### Пример

<code>theresults.in</code> или стандартный ввод	<code>theresults.out</code> или стандартный вывод
2 1	1.00
72 96	1.44
9 2	
3 5 5 5 5 9 11 13 15	
0 0	

## Задача F. Way to Airport

Имя входного файла:	<code>way.in</code> или <i>стандартный ввод</i>
Имя выходного файла:	<code>way.out</code> или <i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На обратном пути участники были доставлены в небольшой прибрежный город, в котором, впрочем, построен международный аэропорт. При этом из-за небольшого шторма теплоход с участниками несколько опоздал, так что водителю автобуса надо доставить участников в аэропорт как можно быстрее.

Система дорог города представляет собой на карте прямоугольную сетку, состоящую из  $N \times M$  перекрёстков. Причал расположен в левом верхнем углу карты, аэропорт — в правом нижнем углу.

Все перекрёстки оборудованы светофорами; правда, от привычных нам светофоров они отличаются: по всем направлениям перекрёстка *одновременно* горит либо зелёный, либо красный свет. Водитель законопослушен и не будет ехать на красный свет, однако из-за отсутствия верхнего предела скорости может проехать расстояние между соседними перекрёстками за нулевое время. Естественно, что водитель знает для каждого светофора все промежутки времени, когда на нём горит красный свет.

Ваша задача — найти минимальный момент времени, в который автобус окажется около аэропорта. Считается, что автобус стартует в нулевой момент времени.

### Формат входных данных

В первой строке входного файла заданы два целых положительных числа  $N$  и  $M$  ( $1 \leq N \times M \leq 1000$ ) — размеры города. В последующих  $N \times M$  строках задан график работы светофоров, перечисленный в порядке обхода карты построчно. Каждое расписание начинается с целого числа  $r_i$  ( $1 \leq r_i \leq 100$ ) — количества промежутков времени, в течение которых на светофоре горит красный свет. Далее в той же строке следует  $r_i$  пар целых положительных чисел, каждое из которых не превосходит  $10^6$  — начало и конец соответствующего отрезка. Отрезки не имеют общих точек и отсортированы по возрастанию момента включения красного света.

### Формат выходных данных

Выходной файл должен содержать единственное целое число — минимальный момент времени, когда автобус сможет приехать в аэропорт.

### Примеры

<code>way.in</code> или стандартный ввод	<code>way.out</code> или стандартный вывод
2 3 2 0 1 4 15 2 0 3 4 15 2 0 1 3 15 1 0 200 1 0 200 1 0 200	15
2 2 1 1 2 1 2 3 1 3 4 1 4 5	0