

## Задача А. Часы

Имя входного файла: a.in  
 Имя выходного файла: a.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 мегабайта

В часах села батарейка, и они стали идти вдвое медленнее. Когда на часах было  $x_1$  часов  $y_1$  минут, правильное время было  $a_1$  часов  $b_1$  минут. Сколько времени будет на самом деле, когда часы в следующий раз покажут  $x_2$  часов  $y_2$  минут.

### Формат входных данных

Заданы числа  $x_1, y_1, a_1, b_1, x_2, y_2$  в указанном порядке. Все числа целые. Числа  $x_1, a_1, x_2$  — от 0 до 23, числа  $y_1, b_1, y_2$  — от 0 до 59.

### Формат выходных данных

Выведите два числа  $a_2, b_2$ , определяющие сколько будет времени на самом деле, когда на часах будет  $x_2$  часов  $y_2$  минут.

### Примеры

| a.in                    | a.out |
|-------------------------|-------|
| 12 34<br>10 34<br>12 35 | 10 36 |
| 12 34<br>10 0<br>2 34   | 14 0  |

## Задача В. Складирование ноутбуков

Имя входного файла: b.in  
 Имя выходного файла: b.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 мегабайта

На склад, который имеет форму прямоугольного параллелепипеда, привезли ноутбуки, упакованные в коробки. Каждая коробка также имеет форму прямоугольного параллелепипеда.

По правилам хранения коробки с ноутбуками должны быть размещены на складе с выполнением следующих двух условий:

- Стороны коробок должны быть параллельны сторонам склада
- Коробку при помещении на склад разрешается расположить где угодно (с выполнением предыдущего условия), в том числе на другой коробке, но все коробки должны быть ориентированы одинаково (т.е. нельзя одну коробку расположить «стоя», а другую – «лежа»)

Напишите программу, которая по размерам склада и размерам коробки с ноутбуком определит максимальное количество ноутбуков, которое может быть размещено на складе.

### Формат входных данных

Вводится шесть натуральных чисел. Первые три задают длину, высоту и ширину склада. Следующие три задают соответственно длину, высоту и ширину коробки с ноутбуком. Каждое из чисел не превышает 1000.

### Формат выходных данных

Выведите одно число — максимальное количество ноутбуков, которое может быть размещено на складе.

### Примеры

| b.in                 | b.out   |
|----------------------|---------|
| 100 200 300<br>1 2 3 | 1000000 |
| 100 200 300<br>3 2 1 | 1000000 |
| 100 100 1<br>2 2 2   | 0       |
| 7 7 7<br>3 3 3       | 8       |

## Задача С. Контрольная по ударениям

|                                           |              |
|-------------------------------------------|--------------|
| Имя входного файла:                       | c.in         |
| Имя выходного файла:                      | c.out        |
| Максимальное время работы на одном тесте: | 1 секунда    |
| Максимальный объем используемой памяти:   | 64 мегабайта |

Учительница задала Пете домашнее задание — в заданном тексте расставить ударения в словах, после чего поручила Васе проверить это домашнее задание. Вася очень плохо знаком с данной темой, поэтому он нашел словарь, в котором указано, как ставятся ударения в словах. К сожалению, в этом словаре присутствуют не все слова. Вася решил, что в словах, которых нет в словаре, он будет считать, что Петя поставил ударения правильно, если в этом слове Петей поставлено ровно одно ударение.

Оказалось, что в некоторых словах ударение может быть поставлено больше, чем одним способом. Вася решил, что в этом случае если то, как Петя поставил ударение, соответствует одному из приведенных в словаре вариантов, он будет засчитывать это как правильную расстановку ударения, а если не соответствует, то как ошибку.

Вам дан словарь, которым пользовался Вася и домашнее задание, сданное Петей. Ваша задача — определить количество ошибок, которое в этом задании насчитает Вася.

### **Формат входных данных**

Вводится сначала число  $N$  — количество слов в словаре ( $0 \leq N \leq 100$ ).

Далее идет  $N$  строк со словами из словаря. Каждое слово состоит не более чем из 30 символов. Все слова состоят из маленьких и заглавных латинских букв. В каждом слове заглавная ровно одна буква — та, на которую попадает ударение. Слова в словаре расположены в алфавитном порядке. Если есть несколько возможностей расстановки ударения в одном и том же слове, то эти варианты в словаре идут в произвольном порядке.

Далее идет упражнение, выполненное Петей. Упражнение представляет собой строку текста, суммарным объемом не более 30000 символов. Строка состоит из слов, которые разделяются между собой ровно одним пробелом. Длина каждого слова не превышает 30 символов. Все слова состоят из маленьких и заглавных латинских букв (заглавными обозначены те буквы, над которыми Петя поставил ударение). Петя мог по ошибке в каком-то слове поставить более одного ударения или не поставить ударения вовсе.

### **Формат выходных данных**

Выведите количество ошибок в Петинем тексте, которые найдет Вася.

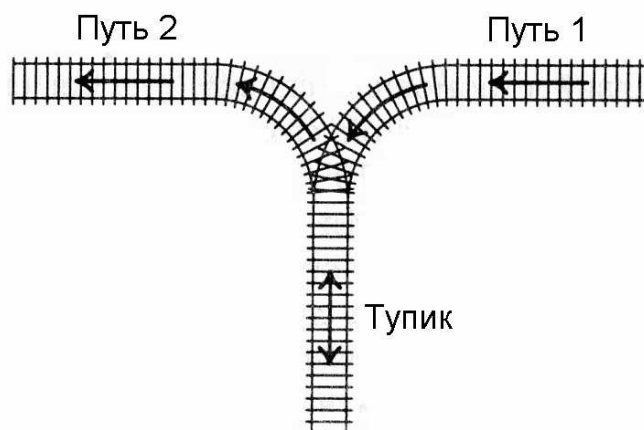
### Примеры

| <b>c.in</b>                                                                         | <b>c.out</b> | <b>Комментарии</b>                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>4<br/> cAnnot<br/> cannOt<br/> fOund<br/> pAge<br/> thE pAge cAnnot be fouNd</p> | <p>2</p>     | <p>В слове cannot, согласно словарю возможно два варианта расстановки ударения. Эти варианты в словаре могут быть перечислены в любом порядке (т.е. как сначала cAnnot, а потом cannOt, так и наоборот).<br/> Две ошибки, совершенные Петей — это слова be (ударение вообще не поставлено) и fouNd (ударение поставлено неверно). Слово thE отсутствует в словаре, но поскольку в нем Петя поставил ровно одно ударение, признается верным.</p> |
| <p>4<br/> cAnnot<br/> cannOt<br/> fOund<br/> pAge<br/> The PAGE cannot be found</p> | <p>4</p>     | <p>Неверно расставлены ударения во всех словах, кроме The (оно отсутствует в словаре, в нем поставлено ровно одно ударение). В остальных словах либо ударные все буквы (в слове PAGE), либо не поставлено ни одного ударения.</p>                                                                                                                                                                                                               |

## Задача D. Сортировка вагонов

Имя входного файла: d.in  
 Имя выходного файла: d.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 мегабайта

К тупику со стороны пути 1 (см. рисунок) подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завести их в тупик (при желании, можно даже завести в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завести в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее



(так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.

Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика). Напишите программу, определяющую, можно ли это сделать.

### Формат входных данных

Вводится число  $N$  — количество вагонов в поезде ( $1 \leq N \leq 100$ ). Далее идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до  $N$ , каждое из которых встречается ровно один раз.

### Формат выходных данных

Если сделать так, чтобы вагоны шли в порядке от 1 до  $N$ , считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите сообщение YES, если это сделать нельзя, выведите NO.

### Примеры

| d.in         | d.out | Комментарии                                                                                                                                                                                             |
|--------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3<br>3 2 1   | YES   | Надо весь поезд завести в тупик, а затем целиком вывезти его на 2-й путь.                                                                                                                               |
| 4<br>4 1 3 2 | YES   | Сначала надо в тупик завести два вагона, один из которых оставить в тупике, а другой — вывезти на 2-й путь, после чего завести в тупик еще два вагона и вывезти 3 вагона, стоящие в тупике, на 2-й путь |
| 3<br>2 3 1   | NO    |                                                                                                                                                                                                         |

## Задача Е. Остаток от деления на цифру

Имя входного файла: e.in  
 Имя выходного файла: e.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 мегабайта

Напишите программу, вычисляющую остаток от деления заданного «длинного» числа на заданную цифру.

### Формат входных данных

В первой строке задана цифра  $K$  ( $1 \leq K \leq 9$ ). Во второй строке задано натуральное число  $N$ , состоящее из не более чем 250 цифр.

### Формат выходных данных

Выведите остаток от деления  $N$  на  $K$ .

### Примеры

| e.in           | e.out |
|----------------|-------|
| 5<br>123456789 | 4     |
| 1<br>123       | 0     |

## Задача F. Заполните массив

Имя входного файла: `f.in`  
 Имя выходного файла: `f.out`  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 мегабайта

Требуется заполнить  $N$  элементов массива, пронумерованных числами от 1 до  $N$  ( $A[1] \dots A[N]$ ), натуральными числами от 2 до  $N+1$ , используя каждое число ровно один раз, так, чтобы значение каждого элемента массива делилось бы нацело на его номер (т.е. для каждого  $i$   $A[i]$  делилось бы на  $i$ ).

Напишите программу, которая для заданного  $N$  заполнит массив согласно описанному правилу.

### Формат входных данных

Вводится одно натуральное число  $N$  ( $1 \leq N \leq 1000$ ).

### Формат выходных данных

Выведите заполненный массив. Если вариантов заполнения несколько, выведите любой из них. Если заполнить массив невозможно, выведите одно число 0.

### Пример

| <code>f.in</code> | <code>f.out</code> |
|-------------------|--------------------|
| 2                 | 3 2                |

## Задача G. Трехцветные таблицы

Имя входного файла: g.in  
 Имя выходного файла: g.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 мегабайта

Прямоугольную таблицу, состоящую из  $N$  строк и  $M$  столбцов, раскрашивают следующим образом. Каждый столбец таблицы и каждую строку красят либо в синий, либо в желтый цвет. В итоге клетки, оказавшиеся на пересечении синего столбца и синей строки оказываются синими, желтого столбца и желтой строки — желтыми, а клетки на пересечении синего столбца и желтой строки, или, наоборот, желтого столбца и синей строки — зелеными.

Раскраска всех клеток таблицы (или просто сама таблица) называется правильной, если она может быть получена описанным выше способом.

Вам дана прямоугольная таблица, которую нужно раскрасить таким образом. Про некоторые клетки известно, какого цвета они должны быть, а остальные клетки могут в итоге быть любого цвета. Определите, как нужно раскрасить остальные клетки таблицы, чтобы раскраска была правильной.

### Формат входных данных

Вводятся числа  $N$  и  $M$  — количество строк и столбцов таблицы ( $1 \leq N \leq 30$ ,  $1 \leq M \leq 30$ ). Далее записано  $N$  строк по  $M$  чисел в каждой, задающие цвета, в которые должны быть окрашены клетки:

- 0 — клетка может в итоге быть любого цвета
- 1 — клетка должна быть синей
- 2 — клетка должна быть желтой
- 3 — клетка должна быть зеленой

### Формат выходных данных

Выведите  $N$  строк по  $M$  чисел в каждой — раскраску таблицы, которая является правильной, и в которой нужные клетки имеют нужный цвет. Если вариантов несколько, выведите любой из них. Если покрасить таким образом таблицу невозможно, выведите одно число 0.

### Примеры

| g.in                                 | g.out                         |
|--------------------------------------|-------------------------------|
| 3 4<br>1 0 0 0<br>3 0 0 0<br>0 0 0 0 | 1 1 1 1<br>3 3 3 3<br>1 1 1 1 |
| 2 2<br>3 3<br>3 3                    | 3 3<br>3 3                    |
| 2 2<br>2 2<br>2 3                    | 0                             |



## Задача Н. Свинки-копилки

Имя входного файла: h.in  
 Имя выходного файла: h.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 мегабайта

У Васи есть  $N$  свинок-копилок, свинки занумерованы числами от 1 до  $N$ . Каждая копилка может быть открыта единственным соответствующим ей ключом или разбита.

Вася положил ключи в некоторые из копилки (он помнит, какой ключ лежит в какой из копилки). Теперь Вася собрался купить машину, а для этого ему нужно достать деньги из всех копилки. При этом он хочет разбить как можно меньшее количество копилки (ведь ему еще нужно копить деньги на квартиру, дачу, вертолет...). Помогите Васе определить, какое минимальное количество копилки нужно разбить.

### Формат входных данных

В первой строке содержится число  $N$  — количество свинок-копилки ( $1 \leq N \leq 100$ ). Далее идет  $N$  строк с описанием того, где лежит ключ от какой копилки: в  $i$ -ой из этих строк записан номер копилки, в которой находится ключ от  $i$ -ой копилки.

### Формат выходных данных

Выведите единственное число: минимальное количество копилки, которые необходимо разбить.

### Пример

| h.in                  | h.out | Комментарий                                                                                                                                                                                      |
|-----------------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4<br>2<br>1<br>2<br>4 | 2     | Ключи от первой и третьей копилки лежат в копилке 2, ключ от второй — в первой, а от четвертой — в ней самой. Чтобы открыть все копилки, достаточно разбить, например, копилки с номерами 1 и 4. |

## Задача I. Медианы объединений

Имя входного файла: `i.in`  
 Имя выходного файла: `i.out`  
 Максимальное время работы на одном тесте: 0.5 секунды  
 Максимальный объем используемой памяти: 64 мегабайта

Дано  $N$  упорядоченных по неубыванию последовательностей целых чисел (т.е. каждый следующий элемент больше либо равен предыдущему), в каждой из последовательностей ровно  $L$  элементов. Для каждой двух последовательностей выполняют следующую операцию: объединяют их элементы (в объединенной последовательности каждое число будет идти столько раз, сколько раз оно встречалось суммарно в объединяемых последовательностях), упорядочивают их по неубыванию и смотрят, какой элемент в этой последовательности из  $2L$  элементов окажется на месте номер  $L$  (этот элемент называют левой медианой).

Напишите программу, которая для каждой пары последовательностей выведет левую медиану их объединения.

### Формат входных данных

Сначала вводятся числа  $N$  и  $L$  ( $2 \leq N \leq 100$ ,  $1 \leq L \leq 300$ ). В следующих  $N$  строках задаются последовательности. Каждая последовательность состоит из  $L$  чисел, по модулю не превышающих 30000.

### Формат выходных данных

В первой строке выведите медиану объединения 1-й и 2-й последовательностей, во второй строке — объединения 1-й и 3-й, и так далее, в  $(N-1)$ -ой строке — объединения 1-й и  $N$ -ой последовательностей, далее медиану объединения 2-й и 3-й, 2-й и 4-й, и т.д. до 2-й и  $N$ -ой, затем 3-й и 4-й и так далее. В последней строке должна быть выведена медиана объединения  $(N-1)$ -й и  $N$ -ой последовательностей.

### Пример

| <code>i.in</code> | <code>i.out</code> |
|-------------------|--------------------|
| 3 6               | 7                  |
| 1 4 7 10 13 16    | 10                 |
| 0 2 5 9 14 20     | 9                  |
| 1 7 16 16 21 22   |                    |