

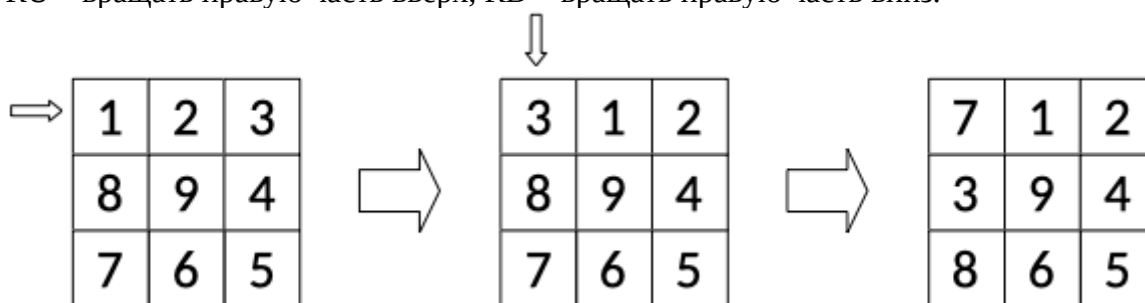
## А. Квадратик Рубика

Ограничение времени: 1 с

Ограничение памяти: 64МБ

Необходимо написать программу, которая найдёт решение головоломки «Квадратик Рубика», названной так из-за своей аналогии с кубиком Рубика. «Квадратик Рубика» – представляет собой одну сторону кубика Рубика, но имеет чуть другие правила перемещения.

Существует всего 8 элементарных действий над головоломкой: UL – вращать верхнюю часть влево, UR – вращать верхнюю часть вправо, DL – вращать нижнюю часть влево, DR – вращать нижнюю часть вправо, LU – вращать левую часть вверх, LD – вращать левую часть вниз, RU – вращать правую часть вверх, RD – вращать правую часть вниз.



На рисунке схематично показан результат разборки головоломки – два последовательных элементарных действия: UR и LD, произведённых над собранной головоломкой.

При вращении любой из частей в одну из сторон, тот пронумерованный квадратик, который расположен на той стороне, в которую вращают часть, оказывается на противоположной стороне, а остальные квадратики смещаются на одно деление (см. рисунок). На рисунке маленькими стрелками показано, как «вращали» стороны квадрата, а большими стрелками указано куда смотреть, чтобы узнать, что получилось в результате элементарного действия.

### Входные данные

Три строки содержащих описание квадрата (см. примеры). Гарантируется, что во входном файле дана несобранная головоломка, которую возможно собрать не более чем за 100 элементарных действий.

### Выходные данные

Одна строка без пробелов, содержащая не более 200 символов, которые описывают набор элементарных действий над «квадратиком Рубика» приводящий к сборке головоломки.

Кубок Псковской области по программированию среди школьников — 2019  
Отборочный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
712 394 865	LUUL
123 894 576	DRDR

## В. Знаки на матрице

Ограничение времени: 2 с

Ограничение памяти: 64МБ

Из-за того, что видеокарты способны очень быстро производить копирование частей видеобуфера прямоугольной формы, может быть целесообразным рассмотреть возможность хранения изображения различных знаков, в том числе различных по величине, используя прямоугольную матрицу пикселей, на которой эти знаки располагаются. Знаки могут располагаться с перекрытием, если имеют одинаковые части, собственно которые и перекрываются. Отдельно хранят список координат и размеры символов.

### Входные данные

Первая строка содержит два числа  $H$  и  $W$  ( $1 \leq H, W \leq 1000$ ) – высоту и ширину матрицы пикселей. Далее идут  $H$  строк, в каждой из которых по  $W$  нулей и единиц, обозначающие выключенные и включенные пиксели соответственно. Затем идёт два числа  $h$  и  $w$  ( $1 \leq h, w \leq 200$ ) – высота и ширина прямоугольного изображения знака. В последующих  $h$  строках идёт описание знака аналогично описанию матрицы пикселей.

### Выходные данные

Два числа – строку и столбец расположения левого верхнего пикселя символа. Если возможных вариантов несколько, выведите любой. Левый верхний пиксель имеет координаты  $(0, 0)$ . Гарантируется, что ответ всегда имеется.

### Примеры

Входные данные	Выходные данные
6 9 000000100 100010010 010010100 000011000 000010100 001010010 5 4 1001 1010 1100 1010 1001	1 4

## С. Сортировка по алфавиту

Ограничение времени: 2 с

Ограничение памяти: 64МБ

Некоторым может не нравиться порядок букв в алфавите, и они меняют его, например, на такой порядок `hizjefgklmynopabcdqrstuvw`. Требуется написать программу, которая имеющиеся списки слов отсортирует в соответствии с новым алфавитом в порядке возрастания.

### Входные данные

Первая строка содержит без пробелов упорядоченный набор символов в соответствии с его порядком в новом алфавите. Далее идёт не более 1000 строк слов (последовательностей строчных латинских букв без пробелов). Гарантируется, что имеется хотя бы одно слово и что общее число символов не превышает одного миллиона.

### Выходные данные

Список всех слов (по одному слову в строке) из входных данных упорядоченный в соответствии с заданием.

### Примеры

Входные данные	Выходные данные
<code>hizjefgklmynopabcdqrstuvw</code>	<code>hello</code>
<code>lamp</code>	<code>zero</code>
<code>hello</code>	<code>know</code>
<code>know</code>	<code>lamp</code>
<code>zero</code>	

## **D. Сортировка пузырьком**

**Ограничение времени:** 1 с

**Ограничение памяти:** 64МБ

Напишите программу, которая отсортирует символы в строке в соответствии с алгоритмом, который описан ниже и выведет процесс сортировки в соответствии с необходимым форматом (см. примеры).

Описание алгоритма сортировки (символы пронумерованы от 1 до N):

```
ЦИКЛ ДЛЯ J=1 ДО N-1 ШАГ 1
  F=0
  ЦИКЛ ДЛЯ I=1 ДО N-J ШАГ 1
    ЕСЛИ A[I] > A[I+1] ТО ОБМЕН A[I],A[I+1] : F=1
  СЛЕДУЮЩЕЕ I
  ЕСЛИ F=0 ТО ВЫХОД ИЗ ЦИКЛА
СЛЕДУЮЩЕЕ J
```

При каждом сравнении символов выведите две строки, в первой – сравниваемые символы, расположенные в текущий момент на своих позициях, во второй строке - саму строку без сравниваемых символов, на их местах – пробелы. После каждого обмена, выведите также две строки, в верхней строке должны быть расположены символы на новых местах (см. примеры).

После того, как процесс сравнения завершён, и сравниваемые знаки оказались на местах, необходимо вывести две строки, первая из которых пустая, а вторая содержит текущее состояние строки. Весь вывод должен быть оформлен в виде таблицы, формат определите по примерам.

### **Входные данные**

Одна строка длиной от 2-х до 10 строчных символов латинского алфавита.

### **Выходные данные**

Выведите процесс сортировки в соответствии с описанным форматом.

Кубок Псковской области по программированию среди школьников — 2019  
 Отборочный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
cab	<pre> +----+  ca      b  +----+  ac      b  +----+        acb  +----+    cb    a    +----+    bc    a    +----+        abc  +----+   ab      c   +----+        abc  +----+ </pre>
ab	<pre> +---+  ab       +---+       ab  +---+ </pre>

## Е. Дата и время в хронологическом порядке

Ограничение времени: 1 с

Ограничение памяти: 64МБ

Достаточно часто дату и время хранят в следующем формате: ГГГГ–ММ–ДД чч:мм:сс. Между датой и временем в точности один пробел. Вам необходимо список имеющихся дат, представленных в указанном формате, расположить в хронологическом порядке.

При этом следует не принимать во внимание даты, которые имеют отличный формат или дата и время имеют недопустимые значения, например, не существует 30 февраля, а 29 февраля не каждый год, не существует 13 месяца и 24 часа (есть нюансы, но для данной задачи не существует) и т.п. Високосный год происходит каждый четвёртый год, не происходит каждый сотый, но в каждый 400 всё-таки существует.

Год должен иметь значение в диапазоне от 1812 до 2120 включительно.

### Входные данные

Список дат, по одной на каждой строке. Гарантируется, что ввод содержит не более 100 строк, хотя бы одна дата и строка соответствует формату «ГГГГ–ММ–ДД чч:мм:сс» и ни одна строка не начинается и не заканчивается пробелом.

### Выходные данные

По одной дате на строку в соответствии с заданием.

### Примеры

Входные данные	Выходные данные
2019-04-01 14:29:01 2018-01-02 03:07:29 2017-17-03 23:01:02 2018-11-29 01:59:59 2018-01-02 03:07:29 2018-01-02 03:07:29	2018-01-02 03:07:29 2018-01-02 03:07:29 2018-11-29 01:59:59
2019-04-01 14:29:01 2018/01/02 03:07:29 17-03-2017 23:01:02 2018-02-29 01:59:59 2020-02-29 01:59:59 2016-02-29 01-59-59 2000-02-29 01:59:59 1600-02-29 01:59:59 Time: 20:00:00 2019-10-23 10:00:00 2019-10-23 10:00:00 Wednesday 2019-10-23 24:00:00	2000-02-29 01:59:59 2019-04-01 14:29:01 2019-10-23 10:00:00 2020-02-29 01:59:59

## **Г. Новое умножение**

**Ограничение времени:** 1 с

**Ограничение памяти:** 64МБ

Василий очень хочет изобрести что-нибудь новое. Начать он решил с умножения. «Умножение Василия» (над окончательным названием он еще думает) для чисел  $A$  и  $B$  имеет следующий вид (в символах обычной математики):

$$A \times B = (A-2019) * (B-2019) + 2019 * (A+B-2019)$$

где 2019 - «Постоянная Василия» (название тоже не окончательное).

Пока Василий трудится над презентацией для своего изобретения, напишите программу, реализующую «Умножение Василия» для двух целых положительных чисел.

### **Входные данные**

Целые положительные числа  $A$  и  $B$  ( $0 < A, B < 1000$ ), разделенные одним пробелом.

### **Выходные данные**

Выведите одно число - результат «Умножения Василия».

### **Примеры**

Входные данные	Выходные данные
47 31	1457



## **Г. Число из остатков**

**Ограничение времени:** 2 с

**Ограничение памяти:** 64МБ

Василий увлёкся одной интересной задачей, которую мы здесь не озвучиваем. При решении этой задачи он разработал алгоритм с применением динамического программирования. Но оказалось, что для выполнения компьютерной программы на его компьютере не хватает памяти.

Василий изменил код программы так, что теперь он прогоняет входные данные дважды и получает не одно число в диапазоне от 0 до 999985999948, а два – остатки от деления искомого числа на 999983 и 1000003.

Он предполагал, что восстановить ответ по его остаткам не составит труда, но он ошибся, полный перебор тратит много времени. Помогите ему в поиске ответа, напишите программу, которая будет справляться с каждой такой задачей не дольше чем за 2 секунды!

### **Входные данные**

Два неотрицательных числа, первое – остаток от деления ответа на 999983, второе – на 1000003.

### **Выходные данные**

Выведите искомый ответ.

### **Примеры**

Входные данные	Выходные данные
12 12	12
19 1000002	1000002
999982 1000002	999985999948