

## Задача В. Кони ходят по очереди

Имя входного файла: knight3.in  
Имя выходного файла: knight3.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке. **Ходы коней должны чередоваться.**

### Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

### Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений. Кони должны ходить по очереди, первым может ходить любой из коней, кони могут сделать различное число ходов.

### Примеры

knight3.in	knight3.out
a1	1 b3
c2	2 a1
c2	1 d4
a1	2 b3
	1 c2
	2 a1

## Задача С. Короля — в угол

Имя входного файла: `king1.in`  
Имя выходного файла: `king1.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

В левом нижнем углу доски  $m \times n$  стоит король. Двое игроков по очереди ходят королем, перемещая его на одну клетку по вертикали вверх, по горизонтали вправо или по диагонали вправо-вверх. Проигрывает тот, кто не может сделать ход. Определите, какой из игроков имеет выигрышную стратегию.

### Формат входных данных

На вход программе подается два натуральных числа  $m$  и  $n$ , не превосходящих  $10^9$ .

### Формат выходных данных

Программа должна вывести номер игрока (1 или 2), который имеет выигрышную стратегию.

### Примеры

<code>king1.in</code>	<code>king1.out</code>
3 4	1

## Задача D. Обход доски

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дана шахматная доска размером  $n \times m$ . Необходимо построить обход всей доски ходом коня так, чтобы конь побывал во всех клетках доски ровно по одному разу. В задачах F1–F6 достаточно построить незамкнутый путь, а в задачах F7–F9 необходимо построить циклический обход, начальная и конечная клетка которого также должны быть связаны ходом коня.

Вы должны сдать на проверку текстовый файл, содержащий  $nm$  строчек. Каждая строчка должна содержать координаты ровно одной клетки. Две соседние координаты должны быть связаны ходом коня и каждая из  $nm$  клеток доски должна встречаться в этом файле ровно один раз. Каждая клетка записывается в виде “a1”, где сначала записывается одна из первых  $n$  букв латинского алфавита, затем — число от 1 до  $m$ . Например, для доски  $4 \times 5$  сданный файл может быть таким:

```
a1
c2
d4
b5
a3
b1
d2
c4
a5
b3
c1
a2
b4
d5
c3
d1
b2
a4
c5
d2
```

### Формат входных данных

Список файлов для сдачи:

- F1: путь на доске  $5 \times 5$ ;
- F2: путь на доске  $7 \times 7$ ;
- F3: путь на доске  $6 \times 12$ ;
- F4: путь на доске  $13 \times 14$ ;
- F5: путь на доске  $20 \times 20$ ;
- F6: путь на доске  $26 \times 26$ ;
- F7: цикл на доске  $6 \times 6$ ;
- F8: цикл на доске  $8 \times 8$ ;
- F9: цикл на доске  $7 \times 16$ .

Кроме этого, сдайте программу, которую вы использовали для нахождения ответов. Эта программа не будет проверяться автоматически. Сдать эту программу необходимо как задачу “F”.

## Задача Е. Равные подстроки

Имя входного файла: eqsubstr.in  
Имя выходного файла: eqsubstr.out  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка  $S = s_1 s_2 \dots s_n$  и множество запросов вида  $(l_1, r_1, l_2, r_2)$ . Для каждого запроса требуется ответить, равны ли подстроки  $s_{l_1} \dots s_{r_1}$  и  $s_{l_2} \dots s_{r_2}$ .

### Формат входных данных

В первой строке записана строка  $S$ , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число  $q$  ( $1 \leq q \leq 100\,000$ ) — количество запросов. В каждой из следующих  $q$  строк записаны числа  $l_1, r_1, l_2, r_2$  ( $1 \leq l_1 \leq r_1 \leq |S|$ ;  $1 \leq l_2 \leq r_2 \leq |S|$ ).

### Формат выходных данных

Для каждого запроса выведите «+», если соответствующие подстроки равны, и «-», в противном случае.

### Примеры

eqsubstr.in	eqsubstr.out
abacaba 4 1 1 7 7 1 3 5 7 3 4 4 5 1 7 1 7	++-+
qa 3 1 1 1 1 2 2 2 2 1 1 2 2	++-

## Задача F. Подстроки

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

Дано  $K$  строк из маленьких латинских букв. Требуется найти их наибольшую общую подстроку.

### Формат входных данных

В первой строке число  $K$  ( $1 \leq K \leq 10$ ). В следующих  $K$  строках — собственно  $K$  строк (длины строк от 1 до 10 000).

### Формат выходных данных

Наибольшая общая подстрока.

### Примеры

<code>substr.in</code>	<code>substr.out</code>
3 abacaba mycabarchive acabistrue	cab

## Задача G. Шифр Бэкона

Имя входного файла: `bacon.in`  
Имя выходного файла: `bacon.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Программисту Васе не повезло — вместо отпуска его послали в командировку на научную конференцию. «Надо повышать уровень знаний», — сказал начальник, «Важная конференция по криптографии, проводится во Франции — а там шифровали еще во времена Ришелье и взламывали чужие шифры еще во времена Виета.»

Вася быстро выяснил, что все луврские картины он уже где-то видел, вид Эйфелевой башни приелся ему еще раньше, чем мышка стерла его с коврика, а такие стеклянные пирамиды у нас делают надо всякими киосками и сомнительными забегаловками. Одним словом, смотреть в Париже оказалось просто не на что, рыбу половить негде, поэтому Васе пришлось посещать доклады на конференции.

Один из докладчиков, в очередной раз пытаясь разгадать шифры Бэкона, выдвинул гипотезу, что ключ к тайнам Бэкона можно подобрать, проанализировав все возможные подстроки произведений Бэкона. «Но их же слишком много!» — вслух удивился Вася. «Нет, не так уж и много!» — закричал докладчик, — «Подсчитайте, и вы сами убедитесь!»

Тем же вечером Вася нашел в интернете полное собрание сочинений Бэкона. Он написал программу, которая переработала тексты в одну длинную строку, выкинув из текстов все пробелы и знаки препинания. И вот теперь Вася весьма озадачен — а как же подсчитать количество различных подстрок этой строки?

### Формат входных данных

На входе дана непустая строка, полученная Васей. Строка состоит только из строчных латинских символов. Ее длина не превосходит 2 000 символов.

### Формат выходных данных

Выведите количество различных подстрок этой строки.

### Примеры

<code>bacon.in</code>	<code>bacon.out</code>
<code>aaba</code>	8

## Задача Н. Сравнение подстрок

Имя входного файла: `substrcmp.in`  
Имя выходного файла: `substrcmp.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка  $S = s_1 s_2 \dots s_n$  и множество запросов вида  $(l_1, r_1, l_2, r_2)$ . Для каждого запроса требуется ответить, какая из подстрок больше —  $s_{l_1} \dots s_{r_1}$  или  $s_{l_2} \dots s_{r_2}$ .

### Формат входных данных

В первой строке записана строка  $S$ , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число  $q$  ( $1 \leq q \leq 100\,000$ ) — количество запросов. В каждой из следующих  $q$  строк записаны числа  $l_1, r_1, l_2, r_2$  ( $1 \leq l_1 \leq r_1 \leq |S|$ ;  $1 \leq l_2 \leq r_2 \leq |S|$ ).

### Формат выходных данных

Для каждого запроса выведите «=», если соответствующие подстроки равны, «>», если первая подстрока больше и «<», если первая подстрока меньше.

### Примеры

substrcmp.in	substrcmp.out
abacaba	=
3	<
1 3 5 7	>
1 3 3 5	
4 7 2 5	
ab	<
2	<
1 1 2 2	
1 1 1 2	

## Задача I. Палиндромы

Имя входного файла: `palindrome.in`  
Имя выходного файла: `palindrome.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, `abba` — палиндром, а `омах` — нет. Для строки  $\alpha$  будем обозначать  $\alpha[i..j]$  ее подстроку длины  $j - i + 1$  с  $i$ -й по  $j$ -ю позицию включительно (позиции нумеруются с единицу). Для заданной строки  $\alpha$  длины  $N$  ( $1 \leq N \leq 100\,000$ ) требуется подсчитать число  $q$  пар  $(i, j)$ ,  $1 \leq i < j \leq n$ , таких что  $\alpha[i..j]$  является палиндромом.

### Формат входных данных

Входной файл содержит одну строку  $\alpha$  длины  $N$ , состоящую из маленьких латинских букв.

### Формат выходных данных

В выходной файл выведите искомое число  $q$ .

### Примеры

<code>palindrome.in</code>	<code>palindrome.out</code>
<code>aaa</code>	3
<code>abba</code>	2
<code>омах</code>	0