

Задача А. Нет, только не строки!!!

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

Вернувшись из отпуска в родную лабораторию НИИ Исследований Данных Строк, Вася обнаружил на столе записку следующего содержания:

Даны четыре строки. Требуется выбрать из каждой по подстроке так, чтобы при приписывании подстроки, выбранной из второй строки, к подстроке, выбранной из первой строки, справа получалась та же строка, что и при приписывании подстроки, выбранной из третьей строки, к подстроке, выбранной из четвертой строки, слева, и при этом суммарная длина всех подстрок была максимально возможной.

Теперь Вася думает, кто же подсунил ему эту записку, и, что более важно, как же все-таки решать эту задачу! Помогите ему!

Формат входного файла

Четыре непустые строки, каждая не более чем из 1000 маленьких латинских букв.

Формат выходного файла

Четыре подстроки, которые необходимо найти. Подстроки могут быть пустыми.

Примеры

<code>nostrings.in</code>	<code>nostrings.out</code>
<code>баса</code>	<code>ба</code>
<code>аба</code>	<code>ба</code>
<code>саба</code>	<code>ба</code>
<code>ба</code>	<code>ба</code>

Задача В. Терминатор

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

Два игрока играют в настольную игру. Игровое поле представляет собой квадратный лабиринт, 8×8 клеток. В некоторых клетках располагаются стенки. Один игрок управляет фишкой-терминатором, а второй — фишкой-беглецом. Игроки ходят по очереди, ходы пропускать нельзя (гарантируется, что ход всегда возможен). За один ход игрок может переместить свою фишку в любую из свободных клеток, расположенных рядом с исходной по горизонтали, вертикали или по диагонали (то есть ходом короля). Терминатор, кроме того, может стрелять в беглеца ракетами. Выстрел идет по прямой в любом направлении по горизонтали, вертикали или диагонали. Если беглец оказывается на линии выстрела

терминатора и не прикрыт стенками, то терминатор незамедлительно делает выстрел (вне зависимости от того, чей ход), и беглец проигрывает. Начальное положение фишек задано. Первый ход делает беглец. Он выигрывает, если сделает ход с восьмой строки за пределы игрового поля, так как остальные границы поля окружены стенками.

Вопрос задачи: может ли беглец выиграть при оптимальной игре обеих сторон?

Формат входного файла

Во входном файле задано игровое поле. Свободная клетка обозначена цифрой 0, а клетка со стенкой — цифрой 1. Клетка, в которой находится беглец, обозначена цифрой 2, а клетка с терминатором — цифрой 3.

Формат выходного файла

В выходной файл выведите число 1, если беглец выигрывает, и -1 — в противном случае.

Примеры

<code>terminator.in</code>	<code>terminator.out</code>
<code>01000000</code>	<code>-1</code>
<code>10100000</code>	
<code>31100000</code>	
<code>00020000</code>	
<code>00000000</code>	
<code>00000000</code>	
<code>00000000</code>	
<code>00000000</code>	

Задача С. И снова сумма...

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с котором разрешается производить следующие операции:

- $add(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $sum(l, r)$ — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Формат входного файла

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «+ i », либо «? l r ». Операция «? l r » задает запрос $sum(l, r)$.

Если операция «+ i » идет во входном файле в начале или после другой операции «+», то она задает операцию $add(i)$. Если же она идет после запроса «?», и результат этого запроса был y , то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Формат выходного файла

Для каждого запроса выведите одно число — ответ на запрос.

Пример

sum.in	sum.out
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	