

Основной тур муниципального этапа ВОШ по информатике 2019

27 ноя 2019, 02:52:10

начало: 28 ноя 2019, 12:00:00

конец: 28 ноя 2019, 16:00:00

длительность: 04:00:00

А. Змеелов Стрелочка

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	snakecatcher.in
Вывод	snakecatcher.out

Змеелов Стрелочка перемещается по прямоугольному полю размером $m \times n$ клеток. Изначально, змеелов расположен в левой верхней клетке и смотрит вниз. Змеелов начинает движение вниз до самой нижней клетки. Если справа еще есть столбцы, то змеелов перемещается на одну клетку вправо и разворачивается на 180 градусов. Теперь он смотрит вверх. По второму столбцу змеелов движется вверх до самой верхней клетки. Если справа есть еще столбцы, то змеелов перемещается на одну клетку вправо и разворачивается на 180 градусов. Теперь он смотрит вниз. И, так далее, змейкой. Перемещаясь по полю, змеелов рисует очередное натуральное число в той клетке, в которой он побывал. В самой верхней левой клетке он рисует число 1. Напишите, пожалуйста, программу, которая будет определять, в какую сторону будет смотреть змеелов Стрелочка, когда он обойдет змейкой все поле и нарисует число $m \cdot n$.

↓	14	15
2	13	16
3	12	17
4	11	18
5	10	19
6	9	20
7	8	21

Формат ввода

В трех строках входного файла snakecatcher.in записаны размеры трех прямоугольных полей для змеелова. В каждой из них по два натуральных числа через пробел m и n , где m - количество строк прямоугольника, а n - количество столбцов. $2 \leq m, n \leq 10^{50}$.

Формат вывода

В файл snakecatcher.out запишите три строки - направления, куда будет смотреть змеелов для каждого из трех полей соответственно. Если он будет смотреть вверх - запишите UP, если вниз - запишите DOWN

Пример

Ввод

7 3

5 5

2 2

Вывод

DOWN

DOWN

UP

Примечания

Вы можете набрать до 40 баллов за решение, которое работает для значений m и n не превышающих 1000000000.

Язык

1

В. Хрустальный куб

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	cube.in
Вывод	cube.out

Три смежных ребра хрустального куба со стороной a лежат на осях координат. Координаты всех вершин куба неотрицательны. Волшебные огоньки допускается размещать в точках куба с целочисленными координатами x, y, z , которые удовлетворяют уравнению $3x + 2y - 4z = 0$.

Напишите, пожалуйста, программу, определяющую максимальное количество волшебных огоньков, которые можно разместить внутри куба, включая его грани и ребра.

Формат ввода

В единственной строке входного файла cube.in записано число a - размер ребра куба. $2 \leq a \leq 2000$

Формат вывода

В файл cube.out запишите одно число - ответ к задаче.

Пример

Ввод

2

Вывод

3

Примечания

В тестовом примере ответ 3, так как огоньки можно разместить в точках со следующими координатами (x, y, z) :

$(0, 0, 0)$

$(0, 2, 1)$

(2, 1, 2)

Вы можете набрать до 80 баллов за решение, которое работает для $a < 310$.

Язык Free Basic 1.04

Набрать здесь

Отправить файл

1

Отправить

С. Простые числа

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	number.in
Вывод	number.out

Простым числом называется натуральное (целое положительное) число, имеющее ровно два различных натуральных делителя — единицу и самого себя. Последовательность простых чисел начинается так: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43... Напишите, пожалуйста, программу, подсчитывающую количество простых чисел из заданного комплекта.

Формат ввода

В первой строке входного файла number.in содержится число n - количество натуральных чисел в комплекте. $1 \leq n \leq 20000$
В следующих n строках записано по одному натуральному числу не превышающему 500000. Все числа комплекта различны.

Формат вывода

В выходной файл number.out следует вывести единственное целое число - ответ к задаче.

Пример

Ввод4
20
3
46
11**Вывод**

2

Примечания

Вы можете набрать до 80 баллов за решение, которое работает для значений n не превышающих 14000.

Вы можете набрать до 60 баллов за решение, которое работает для значений n не превышающих 100.

Язык

1

D. Заклинание

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	spell.in
Вывод	spell.out

В Хогвартсе на уроках магии дети постигают науку формулировки заклинаний. На одном из последних занятий они познакомились с понятием магической силы заклинания. Магическая сила заклинания равна количеству повторений в этом заклинании основного магического сочетания символов. Например, если магическое сочетание "qqq" то заклинание "qqqq" обладает магической силой 2, заклинание "qqqqq" магической силой 0, а заклинание "qqwttqqqjkkookqqqicqq" магической силой 3. Напишите, пожалуйста программу, вычисляющую магическую силу заклинания.

Формат ввода

Первая строка входного файла spell.in соержит основное магическое сочетание - последовательность латинских строчных букв длиной не менее 2 и не более 300 символов. Вторая строка содержит заклинание - последовательность латинских строчных букв длиной не менее 2 и не более 3350000 символов.

Формат вывода

В файл spell.out выведите единственное число - магическую силу заклинания.

Пример

Ввод

ba
baaaabbabar

Вывод

3

Примечания

Вы можете получить до 80 баллов за решение, которое работает для заклинаний длиной не более 1000 символов. Вы можете получить до 60 баллов за решение, которое работает для заклинаний длиной не более 400 символов.

Язык

1

Е. Древообразование

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	treegen.in
Вывод	treegen.out

Дан неориентированный граф G , состоящий из n вершин и m рёбер. Известно, что каждое ребро в графе G соединяет различные вершины, а любую пару вершин соединяет не более одного ребра.

Назовём вершину v *древообразующей*, если после её удаления из графа G (вместе со всеми исходящими из неё рёбрами) остаётся граф $G \setminus v$, который является деревом.

Напомним, что граф называется *деревом*, если он связный (из любой вершины можно добраться до любой другой, двигаясь по рёбрам) и не имеет циклов.

Ваша задача — найти все древообразующие вершины в графе G .

Формат ввода

В первой строке входного файла `treegen.in` даны два целых числа n и m — количество вершин и рёбер соответственно в графе G ($3 \leq n \leq 10^5$, $0 \leq m \leq 10^5$). В i -й из последующих m строк дано описание i -го ребра в формате « $a_i b_i$ », что означает, что i -е ребро соединяет вершины a_i и b_i ($1 \leq a_i, b_i \leq n$). Гарантируется, что каждое ребро в графе G соединяет различные вершины, а любую пару вершин соединяет не более одного ребра.

Формат вывода

В первой строке выходного файла `treegen.out` выведите целое число k — количество древообразующих вершин в графе G . Во второй строке выведите номера этих вершин в порядке возрастания, разделяя их пробелом.

Пример

Ввод**Вывод**

Ввод	Вывод
6 7	2
1 2	3 5
2 3	
6 2	
3 4	
5 3	
4 5	
6 5	

Примечания

Решения, работающие для $n \leq 300$ и $m \leq 300$, будут оценены в 50 баллов.

Решения, работающие для $n \leq 3000$ и $m \leq 3000$, будут оценены в 70 баллов.

Язык

1

Г. Складская задача

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	input.txt
Вывод	output.txt

Гигант интернет-торговли Аззон собирается построить несколько складов для хранения товаров. Склады будут полностью роботизированы (так компания хочет сэкономить на зарплате рабочим), и для управления роботами необходимо написать соответствующее ПО.

Склад имеет n ячеек, пронумерованных числами от 1 до n . Изначально все ячейки пусты. Время от времени происходят события следующих типов:

1. На склад прибывает b_i единиц товара типа a_i . Они размещаются в b_i пустых ячейках склада с наименьшими номерами. Если доступно меньше b_i пустых ячеек, то все они заполняются товаром, а излишки выкидываются.
2. Со склада выгружаются b_i единиц товара типа a_i . При этом они выгружаются из b_i ячеек с наименьшими номерами (после выгрузки ячейки остаются пустыми). Если на складе имеется меньше b_i единиц товара типа a_i — то выгружается всё, что есть.
3. От системы подсчёта статистики приходит запрос — узнать тип товара, который находится в ячейке с номером p_i .

Ваша задача — написать программу, моделирующую работу склада.

Формат ввода

В первой строке входного файла `warehouse.in` даны два целых числа n и q — количество ячеек и количество событий ($1 \leq n, q \leq 10^5$). Далее идут q строк, в которых перечислены события в хронологическом порядке по одному в строке. События первого типа имеют вид « $a_i b_i$ », второго — « $-a_i b_i$ », а третьего — « $0 p_i$ » ($1 \leq a_i, b_i, p_i \leq n, 1 \leq i \leq q$).

Формат вывода

В выходной файл `warehouse.out` выведите тип товара в запрашиваемой ячейке на каждый запрос третьего типа, по одному в строке. Если в запрашиваемой ячейке склада товара нет — выводите «0».

Пример

Ввод

```
10 11
1 3
2 3
-2 1
-1 1
3 4
0 4
-1 1
-2 1
0 5
4 3
0 9
```

Вывод

```
3
0
4
```

Примечания

Решения, работающие для $n \leq 1000$ и $q \leq 1000$, будут оценены в 50 баллов.

Решения, работающие для $n \leq 10^5$, $q \leq 10^5$ и $\sum b_i \leq 10^6$, будут оценены в 70 баллов.

Язык

1