

А. Помощь в шпионаже

Ограничение времени: 1с

Ограничение памяти: 128МБ

Во враждебно настроенной стране к нашему отечеству коммутационное оборудование, используемое в штабе врага, имеет программное управление. Доступ к оборудованию у отечественных разведчиков был вчера и будет завтра. Вчера был скопирован код программного обеспечения оборудования и программисты секретного отдела уже вскрыли код и знают где и что поправить, чтобы в дальнейшем получать информацию из штаба врага, не ставя его в известность об этом.

Но имеется проблема: внедрённые агенты доложили, что имеется в оборудовании аппаратная часть, которая в автоматическом режиме по определённому алгоритму подсчитывает число `CURRENT_HASH` и сверяет его с заранее заданным числом `CONST_HASH`. Если `CURRENT_HASH` не равен `CONST_HASH`, то оборудование отправляют на исследования, на которых выясняют причину неравенства. Будьте уверены, если такое произойдёт, то это провал всей операции и заключение разведчиков в тюрьму с пытками, ведь для врага отечественные разведчики являются иностранными шпионами.

Алгоритм работы аппаратного модуля подсчёта `CURRENT_HASH` следующий:

1. Берётся число из N -го байта дампа памяти программы.
2. Далее вся последовательность значений каждого байта программы рассматривается как сверх большое число, где значение каждого байта является цифрой (значения от 0 до 255 включительно). Следовательно, можно считать, что имеется число записанное в 256-ричной системе счисления.
3. Вперёд этого числа подставляется (приписывается) число полученное на первом шаге.
4. От полученного большого числа берётся остаток по модулю 257, результат вычислений и есть `CURRENT_HASH`.

Входные данные

В первой строке находится одно число `LENGTH` - длина программы в байтах ($1 \leq \text{LENGTH} \leq 2^{20}$).

Во второй находится последовательность чисел (отделённые пробелом друг от друга), каждое из них есть значение, которое хранится в соответствующем байте программы. То есть в ячейке памяти с номером 0 хранится значение, равное первому числу в последовательности, второе число хранится в ячейке с номером 1 и т.д.

Далее с новой строки находится два числа - `CONST_HASH` ($0 \leq \text{CONST_HASH} \leq 256$) и через пробел N ($1 \leq N \leq \text{LENGTH}$).

Выходные данные

Выведите число, которое нужно подставить в программу в ячейку с номером N-1, чтобы «аппаратный» алгоритм получил при подсчёте хэша значение равное CONST_HASH.

Если такое число найти невозможно, выведите «Mission: Impossible» (без кавычек).

Пример

Входные данные	Выходные данные
3 4 1 2 253 2	5

Примечание

Если в качестве второго числа подставить число 5, которое «аппаратный» алгоритм при вычислениях подставит и перед последовательностью, и вычислит $5\ 4\ 5\ 2 \bmod 257$.

Так как число 5 4 5 2 записано в 256-ричной системе счисления, то результат можно вычислить по следующей формуле:

$$(((5 * 256 + 4) * 256 + 5) * 256 + 2) \bmod 257 = 84149506 \bmod 257 = 253$$

В. Рисование окружностей

Ограничение времени: 1с

Ограничение памяти: 128МБ

Алгоритмы, разрабатываемые для компьютеров во второй половине XX века, могут быть вновь востребованы на почве популярности ардуино в начале XXI века. Это связано с тем, что производительность микроконтроллеров, используемых в ардуино при малых рабочих частотах (связано с требованиями к снижению энергопотребления) немногим быстрее микропроцессоров, используемых в компьютерах 80-х годов, отсутствует математический сопроцессор, «умеющий» работать с числами с плавающей точкой на аппаратном уровне, в поделках часто экономически нецелесообразно применение графических ускорителей, и в качестве дисплеев к этим поделкам часто применяются матричные дисплеи с низким разрешением.

Одна из компаний, занимающаяся продвижением ардуино, для повышения конкурентоспособности предлагаемых ею решений агрессивно использует маркетинговый ход, связанный с мощной программной поддержкой.

В одном из графических программных модулей компания использует алгоритм Брезенхема для рисования окружности. Руководство недовольно его производительностью и надеется, что вам удастся реализовать оптимизированную версию алгоритма. Но новая оптимизированная версия должна высвечивать в точности те же пиксели, которые высвечивает и текущая реализация алгоритма.

Входные данные

В первой строке три целых числа отделённых друг от друга пробелом - X , Y , R - координаты центра окружности ($-10^9 \leq X, Y \leq 10^9$) и её радиус ($0 \leq R \leq 10^9$).

Во второй строке два целых положительных числа W (количество пикселей по оси X) и H (количество пикселей по оси Y), размеры дисплеев таковы, что количество пикселей на дисплее не превосходит величины в 10^5 . Координата $(0, 0)$ соответствует пикселю, расположенному в левом верхнем углу экрана, а $(W-1, H-1)$ - в правом нижнем.

Выходные данные

Изображение (см. примеры).

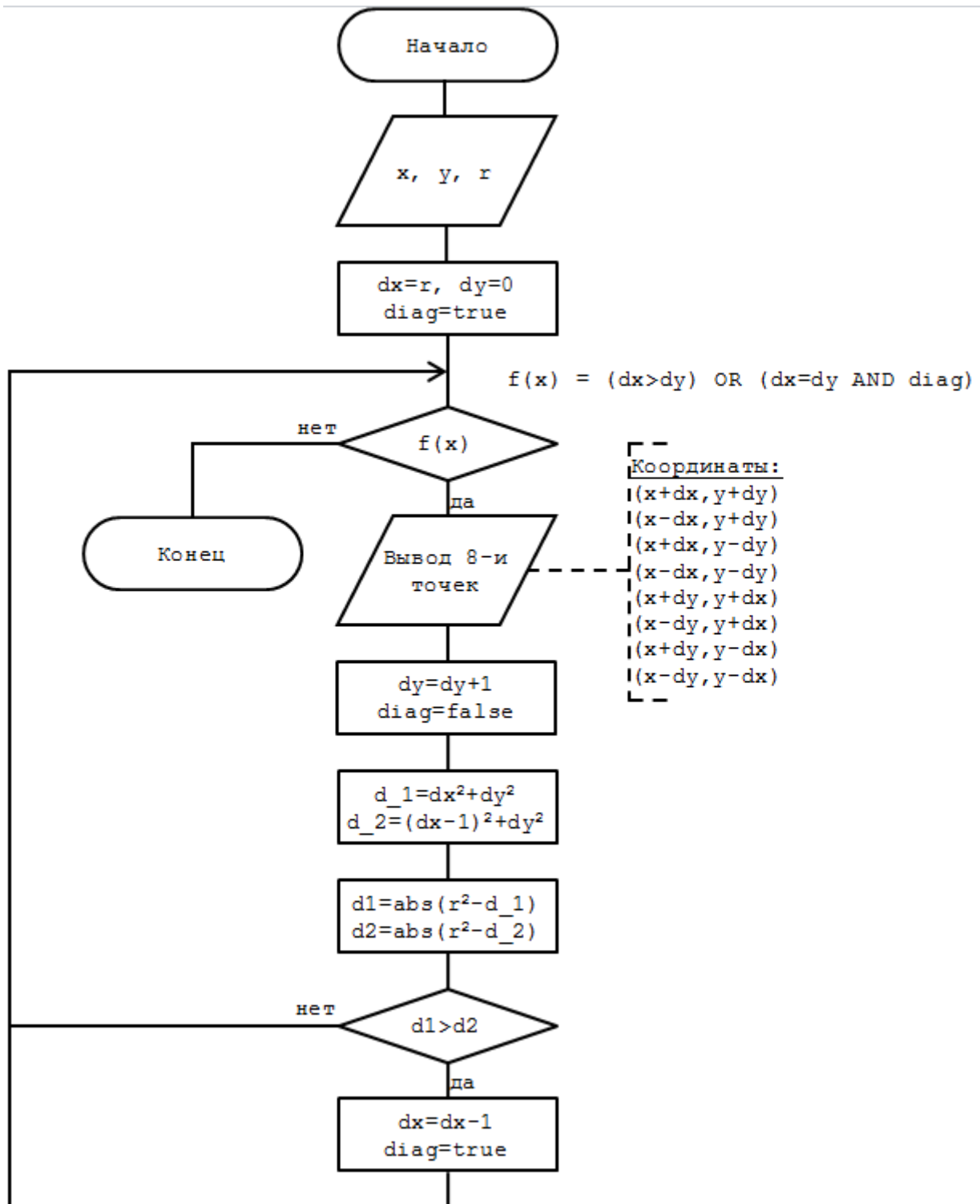
Кубок Псковской области по программированию среди школьников - 2017
 Финальный тур, 7 ноября 2017 года

Пример

Входные данные	Выходные данные
4 6 5 16 8	<pre> +-----+ ##### # # # # # # # # # +-----+ </pre>
5 5 4 6 6	<pre> +-----+ ## # # # # # +-----+ </pre>
5 6 6 4 5	<pre> +-----+ # # # # +-----+ </pre>
0 0 0 1 1	<pre> +--+ # +--+ </pre>
0 0 1 1 1	<pre> +--+ +--+ </pre>

Примечание

Реализация алгоритма Брезенхема, используемая компанией (блок-схема):



С. Склад

Ограничение времени: 1с

Ограничение памяти: 128МБ

Василий разрабатывает программу для автоматизации склада.

На склад приходят и уходят разные товары. Нужно определить, сколько и каких товаров осталось на складе после всех операций. Изначально склад пустой. Все операции корректные, т.е. со склада не забирается товара больше, чем в данный момент содержится.

Входные данные

В первой строке целое число N ($1 \leq N \leq 1000$) - количество операций. Далее в N строках приведены операции. Каждая операция описывается типом, количеством и наименованием, разделенным одним пробелом.

- Тип - «IN», если товар пришел на склад и «OUT», если ушел (без кавычек).
- Количество - целое положительное число, не превышающее 1000 - количество товара.
- Наименование - строка, состоящая из заглавных и строчных латинских букв и пробелов, длиной не менее одного и не более 100 символов. Гарантируется, что наименование не начинается и не заканчивается пробелом.

Выходные данные

Для каждого из оставшихся товаров на складе выведите его количество и наименование, разделенные одним пробелом. Товары отсортируйте по наименованию лексикографически.

Если на складе товаров не осталось, выведите «-1» (без кавычек).

Пример

Входные данные	Выходные данные
3 IN 10 Keyboard IN 5 Mouse OUT 5 Keyboard	5 Keyboard 5 Mouse
3 IN 10 Keyboard OUT 7 Keyboard OUT 3 Keyboard	-1
1 IN 10 Name With Space	10 Name With Space

D. Простое счастье

Ограничение времени: 1с

Ограничение памяти: 128МБ

Василию редко попадаются счастливые билеты. Поэтому он решил придумать свое определение счастливого билета.

Он называет билет счастливым, если его номер можно разбить на простые числа. Числа в разбиении могут иметь ведущие нули, т.е. число 005 будет также считаться простым.

Помогите Василию определить, является ли билет счастливым.

Входные данные

Дано одно целое число, состоящее ровно из 6 цифр. Число может содержать ведущие нули.

Выходные данные

Если билет является счастливым по определению Василия, выведите «YES», иначе выведите «NO» (без кавычек).

Пример

Входные данные	Выходные данные
611729	YES
002458	NO

Е. Сортировка таблицы

Ограничение времени: 1с

Ограничение памяти: 128МБ

Василий разрабатывает новый текстовый редактор. Его главной особенностью будет возможность работать с таблицами в текстовом виде, в том числе сортировать их.

Отображение таблиц Василий уже сделал, помогите сделать сортировку.

Входные данные

В первой строке дано целое число - номер столбца, по которому нужно отсортировать таблицу и направление сортировки: «ASC» - лексикографически по возрастанию, «DESC» - лексикографически по убыванию.

В следующих строках дана таблица. Формат см. в примере ввода.

Значения в ячейках выровнены по левому краю.

Количество столбцов не превышает 10.

Количество строк не превышает 100.

Ширина ячейки не превышает 10 символов (не включая границы).

Текст в ячейках таблицы состоит из заглавных и строчных латинских букв, цифр и пробелов и не начинается с пробела.

Гарантируется, что в таблице существует столбец с данным номером, а также то, что значения ячеек в указанном столбце различное.

Выходные данные

Выведите таблицу в том же формате, что и во входных данных, строки которой отсортированы по значениям указанного столбца. Сохраните ширины столбцов.

Пример

Входные данные	Выходные данные
2 ASC +-----+-----+-----+ hello world 111 +-----+-----+-----+ test table 222 +-----+-----+-----+	+-----+-----+-----+ test table 222 +-----+-----+-----+ hello world 111 +-----+-----+-----+

Ф. Матрёшки

Ограничение времени: 1с

Ограничение памяти: 128МБ

У Василия есть коллекция матрёшек. Матрёшки можно вкладывать друг в друга: в матрёшку большего диаметра можно вложить ровно одну матрёшку меньшего диаметра. Помогите Василию выяснить, какое минимальное количество матрёшек можно получить из имеющихся.

Входные данные

В первой строке дано целое число N ($1 \leq N \leq 1000$). Далее в N строках даны целые числа A_i ($1 \leq A_i \leq 1000$) - диаметры матрёшек.

Выходные данные

Выведите одно целое число - минимальное количество матрёшек, которое можно получить из данных.

Пример

Входные данные	Выходные данные
5 3 1 4 1 2	2

Г. Очередь

Ограничение времени: 1с
Ограничение памяти: 128МБ

В очереди N посетителей, Василий спросил каждого посетителя, за кем тот занимал очередь, ему либо называли номер посетителя, либо отвечали, что они «только спросить». Василий заметил, что первым проходит тот, кому «только спросить» с наименьшим номером, далее проходят все, стоящие за ним. Если несколько человек занимали очередь за одним и тем же посетителем, то сначала проходит тот, у кого номер меньше (и все, стоящие за ним).

Номер Василия $N+1$ и он хочет пройти как можно быстрее. Он может либо занять очередь за посетителем, за которым больше никто не занимал, либо сам стать тем, кому «только спросить». Если у Василия есть равноценный выбор, занимать очередь или «только спросить», он предпочтет занять очередь.

Время обслуживания каждого посетителя одинаковое.

Входные данные

В первой строке дано целое число N ($2 \leq N \leq 10000$).

Во второй строке через один пробел даны ответы посетителей в очереди - либо число a_i ($1 \leq a_i \leq N$, $a_i \neq i$), либо знак «!», если посетителю «только спросить».

Выходные данные

Выведите либо номер посетителя, за которым нужно занять очередь, либо знак «!», если Василий должен «только спросить».

Пример

Входные данные	Выходные данные
10 2 ! 1 ! 3 5 4 3 10 !	6
2 2 1	!

Н. Робот в столовой

Ограничение времени: 1с

Ограничение памяти: 128МБ

Василий программирует робота для автоматизации столовой. Сейчас он учит робота мыть посуду.

Каждый посетитель столовой берет одну тарелку, одну кружку, одну ложку или вилку (всего три предмета).

Робот для каждого предмета выполняет следующие действия:

1. Берет предмет
2. Моет
3. Сушит
4. Переносит к чистой посуде (соответственно типу предмета)

Вам дано количество посетителей столовой за сегодня. Вычислите, какое количество действий совершил робот Василия.

Входные данные

Одно целое число N ($1 \leq N \leq 100$) - количество посетителей столовой.

Выходные данные

Целое число - количество действий.

Пример

Входные данные	Выходные данные
3	36