

А. Турникет на КПП

Ограничение времени	2 секунды
Ограничение памяти	64.0 Мб

На КПП в кадетский корпус стоит турникет. Чтобы через него пройти, требуется приложить пропуск. Пропуск надо прикладывать и при входе через КПП и при выходе. Для того, чтобы исключить несанкционированные проходы, пропуск не работает два раза подряд как на вход, так и на выход.

Иногда кадеты забывают взять с собой пропуск, поэтому хитрые кадеты придумали, как по одному пропуску проходить по несколько человек, чтобы не возвращаться обратно. Чтобы войти или выйти вдвоем по одному пропуску, кадеты прикладывают его с нужной стороны турникета, потом с противоположной, но никто не проходит, а затем снова с нужной. Старший воспитатель решил разобраться с данной проблемой и сделать выговоры всем нарушителям. По каждому событию входа/выхода есть запись в журнале событий. Он считает нарушителями тех владельцев пропусков, у которых произошло три события вида выход-вход-выход менее чем за t минут.

Вам дан журнал событий турникета. Требуется вывести список тех кадет, которым будет сделан выговор.

Формат ввода

В первой строке задано два числа n ($1 \leq n \leq 1000$) и t ($3 \leq t \leq 1440$) — число записей в журнале событий турникета и ограничение времени, выбранное начальником охраны, соответственно.

В следующих n строках даны записи в журнале событий в хронологическом порядке. Запись в журнале состоит из трех частей, разделенных пробелом:

- Время события в формате $hh:mm$
- Фамилия кадета, состоящая из не более чем 20 букв латинского алфавита, первая из которых заглавная.
- Тип события: *in*, если произошел вход и *out*, если произошел выход. Гарантируется, что не существует двух событий, которые происходят одновременно. Также гарантируется, что у любых двух разных кадетов разные фамилии и у одного кадета не бывает двух событий одного типа подряд.

Формат вывода

В первой строке выведите число нарушителей. После чего выведите фамилии нарушителей в лексикографическом порядке.

Ввод	Вывод
<pre>6 10 02:35 Orlov in 02:36 Sorokin out 02:37 Orlov out 02:39 Sorokin in 02:44 Orlov in 02:45 Sorokin out</pre>	<pre>1 Sorokin</pre>
<pre>6 10 06:44 Orlov in 06:45 Sorokin out 06:46 Orlov out 06:48 Sorokin in 06:54 Orlov in 06:55 Sorokin out</pre>	<pre>0</pre>

В. Рацион кадета

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб

У завпроизводства столовой кадетского корпуса возникла проблема: кадеты начали жаловаться на однообразное меню. Узнав, что кадет Степан хорош в программировании, завпроизводством попросил его написать программу, которая помогала бы с распределением блюд.

Завпроизводства считает, что в качестве показателя того, насколько меню кадет однообразно, можно взять число одинаковых блюд, которые даются в последовательные приемы пищи. Назовем этот показатель «индекс единообразия меню». То есть, если представить расписание блюд как массив $menu$, то индекс единообразия меню будет равен количеству индексов i таких, что $menu_i = menu_{i-1}$. Для начала вас просят найти не сам порядок блюд в меню, а хотя бы минимально возможное значение индекса единообразия меню, которое можно было бы получить некоторой перестановкой заданного набора блюд. Помогите кадетской столовой!

Формат ввода

В первой строке содержится число n — количество блюд, которые должны войти в меню ($1 \leq n \leq 100$). В следующей строке содержится n чисел a_i — блюда ($1 \leq a_i \leq 100$). Одинаковые блюда обозначены одинаковыми числами, разные — разными.

Формат вывода

В единственной строке выведите одно число — минимальное возможное значение «индекса единообразия меню».

Ввод	Вывод
5 1 1 3 2 1	0
6 1 1 1 3 1 2	1

Примечания

В первом тестовом примере можно составить следующее расписание блюд: 1,2,1,3,1 тогда «индекс единообразия меню» будет равен 0.

Во втором тестовом примере «индекс единообразия меню», равный 0, достичь не получится, поэтому наименьшая возможная повторяемость равна 1: 1,2,1,3,1,1.

С. Посадочная площадка

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб

Дана карта участка местности, разбитого на квадраты (n строк и m столбцов). На этом участке проходят тренировки (имитация боевых действий) и нужно выбрать безопасное место для посадки вертолѐта, имеющее форму квадрата. Квадраты, непригодные для посадки, обозначены на карте участка нулями, а пригодные – единицами. Так как во время боевых действий обстановка меняется стремительно, необходимо очень быстро находить наибольшее возможное место для посадки. Вас просят написать программу, которая бы определяла наибольший размер площадки, на которой можно совершить посадку вертолѐта.

Формат ввода

Первая строка содержит два натуральных числа: размеры участка n и m ($1 \leq n, m \leq 1000$). Следующие n строк содержат построчную карту местности, где области, пригодные для посадки вертолѐта, обозначены единицами, а остальные – нулями.

Формат вывода

Вывести одно число – длину стороны наибольшей квадратной области пригодной для посадки вертолѐта.

Ввод	Вывод
<pre> 6 7 1111111 1111110 1011110 0011110 1000010 1100111 </pre>	<pre> 4 </pre>

D. Регламент ИнфоТехКвеста

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб

В следующем году регламент олимпиады ИнфоТехКвест немного поменяется. Правила отбора в финальный этап будут следующими:

1. Призеры финала ИнфоТехКвеста прошлого года приглашаются на финал вне зависимости от результата, который они получили на отборочном этапе.
2. Все участники, набравшие на отборочном этапе не меньше баллов, чем установленный жюри пороговый балл, проходят в финальный этап ИнфоТехКвеста.
3. Если в некотором образовательном учреждении ни один участник по первым двум правилам в финальный этап не прошел, то на финальный этап приглашается один участник из этого учреждения, набравший в нём максимальный балл (это не касается учреждений, от которых участников не было).
4. На финальный этап ИнфоТехКвеста можно пригласить не более m участников.

Известно, что никакие два участника ИнфоТехКвеста по итогам первого этапа не набрали одинаковое количество баллов. По информации о результатах отборочного этапа помогите жюри установить минимально возможный пороговый балл, при котором все правила отбора будут выполнены.

Формат ввода

В первой строке содержатся три целых числа n , m и r - число участников отборочного этапа, максимально возможное число участников финального этапа и число образовательных учреждений, из которых могли быть участники ($1 \leq m < n$). Далее в n строках содержатся результаты каждого из участников. Каждая строка состоит из четырех целых чисел. Сначала идет id - уникальный идентификатор участника ($1 \leq id \leq n$), далее $school$ - номер учреждения, в которой данный участник обучается ($1 \leq school \leq r$), затем $score$ - число баллов, набранных участником, четвертое число равно 1, если участник является призером олимпиады прошлого года, и 0 - в противном случае. Гарантируется, что все идентификаторы участников различны, никакие два участника не набрали одинаковое число баллов, и выполнить все правила отбора возможно.

Формат вывода

Выведите одно число – минимальный пороговый балл, который можно установить.

Ввод	Вывод
9 6 5 6 1 799 0 2 4 995 0 1 4 989 1 7 2 538 0 5 4 984 0 8 2 1000 0 3 2 998 0 4 2 823 1 9 1 543 0	985

Е. Расписание нарядов

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб

В некоторой Военной Академии каждый день некоторое количество курсантов заступают в наряд, чтобы охранять территорию. В целях безопасности в каждый момент времени в наряде должен быть хотя бы один курсант. Также, чтобы не отвлекать курсантов от учёбы, администрация Академии постановила, что расписание нарядов должно быть составлено так, чтобы в наряд каждый день заходило как можно меньшее количество курсантов. Определять, что в наряде участвует минимально необходимое количество курсантов, решили по следующему правилу: удаление любого из курсантов из расписания нарядов приводит к появлению промежутка времени, когда в наряде нет ни одного из курсантов. Администрация Академии для удобства составления расписания нарядов поделила сутки на 10000 равных промежутков времени. Известно, что всего в течении дня в наряд заступают n курсантов. От Вас не требуется составить график захода в наряд курсантов, а лишь определить по данному расписанию, удовлетворяет ли оно заданным требованиям. У каждого курсанта известны моменты времени заступления в наряд и окончания наряда s и t . В момент времени s считается, что курсант уже находится в наряде, а в момент времени t наряд уже окончен.

Формат ввода

В первой строке дано единственное целое положительное число n ($1 \leq n \leq 10^5$) – количество курсантов, заступающих в конкретный день в наряд. В каждой из последующих n строк содержится два целых неотрицательных числа s_i и t_i ($a \leq s_i < t_i \leq b$) – время заступления в наряд и окончания наряда i -го курсанта.

Формат вывода

В единственной строке выходного потока необходимо вывести «YES», если расписание удовлетворяет требованиям, «NO» - в противном случае.

Ввод	Вывод
4 0 3333 3333 6666 5833 9166 7500 10000	YES
4 0 3333 4166 6666 5833 9166 7500 10000	NO