

## Задача А. Время прилёта

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

При полётах на самолетах в качестве времени вылета и прилёта используется местное время аэропортов вылета и прилёта.

Часовые пояса характеризуются разницей во времени с меридианом, на котором расположена Гринвичская обсерватория. Для каждого часового пояса вводится отклонение от UTC (Всемирного координированного времени).

Например, Москва расположена в часовом поясе UTC+3, а Новосибирск — в часовом поясе UTC+7. Если вылететь из Москвы рейсом в 11:15 и временем полёта ровно в 4 часа, то прилёт будет в Новосибирск будет в 19:15 (4 часа полёта и 4 часа разницы во времени).

Часовые пояса могут изменяться от UTC-11 (Американское Самоа) до UTC+14 (острова Лайн, Кирибати).

По заданному времени вылета и времени полёта, а также по часовым поясам аэропортов вылета и прилёта, вам необходимо определить местное время прилёта и количество дней, прошедших в пути.

### Формат входных данных

В первой строке записаны целые числа  $H_D, M_D$  ( $0 \leq H_D \leq 23, 0 \leq M_D \leq 59$ ) — время вылета.

Во второй строке записаны целые числа  $H_F, M_F$  ( $0 \leq H_F \leq 10^9, 0 \leq M_F \leq 59$ ) — время полёта.

В третьей строке записаны целые числа  $D, A$  ( $-11 \leq D, A \leq 14$ ) — часовые пояса аэропорта вылета и прилёта.

### Формат выходных данных

Выведите три числа  $H_A, M_A, Days$  — время прилёта в часах и минутах, а также разницу в датах между датой вылета и датой прилёта.

### Система оценки

Решения, верно работающие для рейсов, дата вылета и прилёта которых не отличаются, будут набирать не менее половины баллов.

### Пример

стандартный ввод	стандартный вывод
11 15 4 0 3 7	19 15 0
12 0 1 0 -10 13	14 0 1

### Замечание

Первый тест соответсвуете разобранному в условии примеру с Москвой и Новосибирском.

Второй тест соответствует, например, часовому перелету из Американского Самоа на Самоа. Самолет вылетает в 12:00, летит в течение часа и приземляется в 14:00 местного времени. Т.к. он пересёк линию перемены даты, то на Самоа уже наступил следующий день.

В реальности существуют часовые пояса, которые отличаются от UTC на нецелое число часов, однако в задаче они не рассматриваются.

## Задача В. Двумерный дождь

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Игра PitCraft происходит в двумерном мире, который состоит из блоков размером 1 на 1 метр. Остров игрока представляет собой набор столбцов различной высоты, состоящих из блоков камня и окруженный морем.

Над островом прошёл сильный дождь, который заполнил водой все низины, а не поместившаяся в них вода стекла в море, не увеличив его уровень. По ландшафту острова определите, сколько блоков воды осталось после дождя в низинах на острове.

### Формат входных данных

В первой строке записано натуральное число  $N$  ( $0 \leq N \leq 100\,000$ ) — количество столбцов, задающих ландшафт острова.

Во второй строке записано  $N$  натуральных чисел  $H_i$  ( $1 \leq H_i \leq 10^9$ ) — высоты столбцов.

### Формат выходных данных

Выведите одно число — количество блоков занятых водой.

### Система оценки

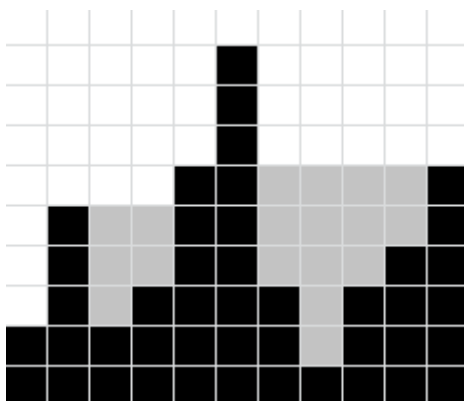
Решения, верно работающие при  $N \leq 100$ , будут набирать не менее половины баллов.

### Пример

стандартный ввод	стандартный вывод
11 2 5 2 3 6 9 3 1 3 4 6	18

### Замечание

Пример соответствует рисунку. Черным цветом обозначен камень, серым — вода.



## Задача С. Падающие домино

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася обожает выставлять сложные фигуры из костяшек домино и, толкнув одну из них, смотреть, как вся конструкция падает. Однако, он сделал уже так много фигур, что решил придумать что-то новое.

Для своей новой идеи он использует костяшки не только длиной 2, но и более длинные (и более короткие). Все костяшки выстраиваются в одну линию на расстоянии 1, а цель игры — опрокинуть все костяшки толкнув наименьшее количество костяшек.

Каждую костяшку можно толкнуть влево или вправо, падая она опрокидывает все костяшки, находящиеся на расстоянии строго меньшем высоты падающей костяшки. При этом те костяшки, которые упали в результате падения на них других костяшек также падают в ту же сторону и, в свою очередь, могут опрокидывать и другие костяшки и так далее.

### Формат входных данных

В первой строке записано натуральное число  $N$  ( $0 \leq N \leq 1\,000\,000$ ) — количество костяшек.

Во второй строке записано  $N$  натуральных чисел  $H_i$  ( $1 \leq H_i \leq 1\,000\,000$ ) — высоты костяшек.

### Формат выходных данных

Выведите число  $M$  — наименьшее количество костяшек, которые нужно толкнуть, чтобы вся конструкция упала.

В следующих  $M$  строках выведите описание костяшек, которые необходимо толкнуть: номер костяшки (нумерация начинается с единицы и идет слева-направо), а также направление толчка: букву "L" для толчка влево и "R" для толчка вправо. Номер костяшки и букву разделяйте пробелом.

Порядок вывода костяшек, которые нужно толкнуть, может быть произвольным. Если решений несколько — выведите любое из них

### Система оценки

Решения, верно работающие при  $N \leq 1000$ , будут набирать не менее половины баллов.

### Пример

стандартный ввод	стандартный вывод
6 1 2 1 4 1 3	1 6 L
7 1 2 4 1 2 3 2	2 3 R 2 L

### Замечание

В первом примере последняя костяшка толкается влево, опрокидывая костяшки с номерами 4 и 5 (их высоты 4 и 1 соответственно). Костяшка номер 4 также падает налево и опрокидывает костяшки с номерами 1, 2 и 3.

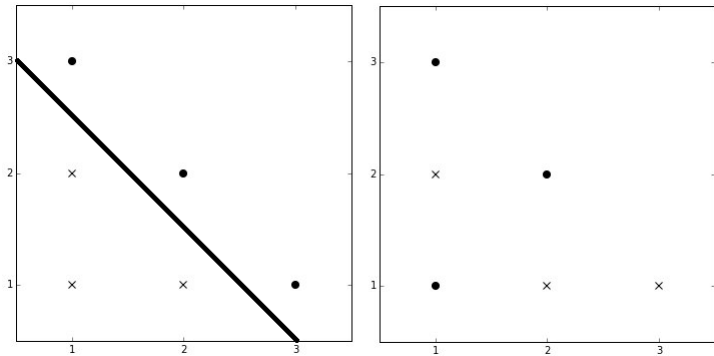
Во втором примере костяшка номер 3 толкается вправо, опрокидывая костяшки номер 4, 5 и 6. Костяшка номер 6 также падает вправо и опрокидывает костяшку номер 7. После этого костяшка номер 2 толкается влево и опрокидывает костяшку номер 1.



## Задача Е. Линейная классификация

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В машинном обучении часто возникает задача линейной классификации объектов, когда классы объектов разделяются между собой линейной поверхностью. Например, у нас есть информация о количестве дней с момента регистрации аккаунта в социальной сети и количество отправленных сообщений за последний день, а также информация о том, является ли этот аккаунт спам-ботом. Возраст аккаунта мы можем взять за  $X$  координату точки, а количество сообщений — за  $Y$  координату. Задача классификации состоит в том, чтобы провести какую-либо прямую так, чтобы объекты одного типа находились по одну сторону этой прямой, а объекты другого типа — по другую.



При наличии такой прямой мы сможем прогнозировать тип даже незнакомого объекта по известному возрасту аккаунта и количеству отправленных сообщений в зависимости от того, с какой стороны от прямой оказался объект. Естественно, в реальных данных могут быть ошибки измерений или необычные объекты и провести такую прямую не всегда возможно, потому что, например, объект первого типа может случайно попасть в скопление объектов второго типа и отделить его прямой невозможно.

Вам необходимо по информации о параметрах и типе объектов определить, существует ли прямая, которая однозначно разделит классы объектов. Прямая не должна проходить ни через один объект.

### Формат входных данных

В этой задаче входной файл содержит несколько тестовых блоков.

В первой строке задано число  $T$  — количество тестовых блоков ( $1 \leq T \leq 100$ ).

Каждый тестовый блок состоит из числа  $N$  — количество описанных объектов ( $1 \leq N \leq 2000$ ).

В следующих  $N$  строках содержится описания объектов, состоящие из трех целых чисел  $X, Y, Type$  ( $0 \leq X, Y \leq 10^7, 0 \leq Type \leq 1$ ).

### Формат выходных данных

Выведите  $T$  слов «YES» или «NO» по одному в строке для каждого из тестовых блоков. «YES» необходимо выводить если разделение на классы возможно, «NO» — если невозможно.

### Система оценки

Решения, верно работающие при  $T \leq 10, N \leq 100$ , будут набирать не менее половины баллов.

## Пример

стандартный ввод	стандартный вывод
2	YES
6	NO
1 1 1	
1 2 1	
1 3 0	
2 1 1	
2 2 0	
3 1 0	
6	
1 3 0	
2 2 0	
1 2 1	
3 1 1	
2 1 1	
1 1 0	