

## Задача 1. Речные прогулки

Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Вдоль течения реки размещены  $n$  пристаней, пронумерованных числами от 1 до  $n$ . Пристань номер 1 находится выше всех остальных по течению реки, пристань номер  $n$  находится в устье реки, расстояние между соседними пристанями равно 1 км.

Для развития туризма решено открыть два прогулочных речных маршрута. Маршруты будут начинаться на одной из промежуточных пристаней (пристани номер 1 или  $n$  не могут быть начальными точками маршрутов), один маршрут будет идти вверх по течению реки к пристани номер 1, другой маршрут будет идти вниз по течению к пристани номер  $n$ . Промежуточных остановок на маршрутах нет.

Для подъёма вверх по течению реки судно тратит  $a$  минут на один километр, а для спуска вниз по течению реки —  $b$  минут на один километр. Определите, на какой пристани должны начинаться оба маршрута, чтобы их продолжительности различались как можно меньше. Это значит, что необходимо минимизировать модуль разности времени в пути двух маршрутов.

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  ( $3 \leq n \leq 2 \cdot 10^9$ ) — общее количество пристаней на маршруте. Вторая строка содержит число  $a$  — время подъёма судна на один километр вверх по течению реки, третья строка содержит число  $b$  — время спуска на один километр вниз по течению,  $1 \leq b < a \leq 2 \cdot 10^9$ .

### Формат выходных данных

Программа должна вывести одно число — номер пристани, на которой необходимо организовать начальный пункт маршрутов. Если возможных подходящих ответов несколько, можно вывести любой из них.

### Система оценки

Решения, правильно работающие, когда все входные числа не превосходят 100, будут оцениваться в 60 баллов.

### Пример

стандартный ввод	стандартный вывод
8	3
7	
3	

### Замечание

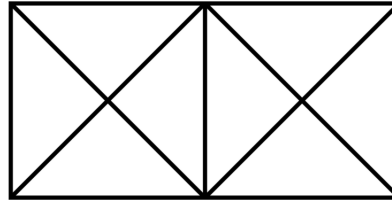
В примере из условия начальным пунктом маршрутов нужно сделать пристань 3. Тогда вверх по течению судно поднимется за  $(3 - 1) \times 7 = 14$  минут, а вниз по течению реки спустится за  $(8 - 3) \times 3 = 15$  минут. Разница в продолжительности маршрутов составит 1, меньшей разности в данном примере достичь невозможно.

## Задача 2. Треугольники

Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Дана клетчатая сетка, состоящая из  $n \times m$  клеток со стороной 1, в каждой клетке проведены обе диагонали.

Например, сетка  $1 \times 2$  выглядит следующим образом:



Назовём прямоугольник на данной сетке *подходящим*, если его вершины расположены в узлах сетки, а длины его стороны равны 1 или 2 (то есть подходящими являются прямоугольники  $1 \times 1$ ,  $1 \times 2$ ,  $2 \times 1$ ,  $2 \times 2$ ).

Треугольник называется *хорошим*, если его стороны образованы сторонами и/или диагоналями сетки и он целиком лежит в каком-то *подходящем* прямоугольнике.

Посчитайте количество *хороших* треугольников на данной сетке.

### Формат входных данных

Программа получает на вход два числа  $n$  и  $m$ , записанных в отдельных строках, — размеры сетки,  $1 \leq n \leq 10^8$ ,  $1 \leq m \leq 10^8$ .

### Формат выходных данных

Программа должна вывести одно целое число — количество искомым треугольников.

Обратите внимание на то, что ответ в этой задаче может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `long long` в языке C++, тип `int64` в Pascal, тип `long` в Java и C#).

### Система оценки

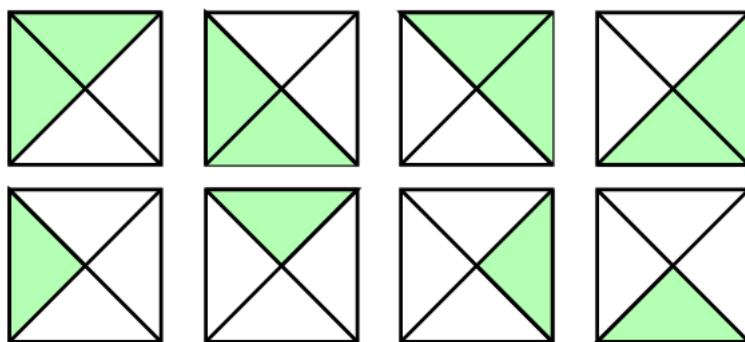
В данной задаче 20 тестов помимо тестов из условия, каждый из них оценивается в 5 баллов. При этом в 4 тестах (помимо тестов из условия)  $n$  или  $m$  равно 1, в 4 других тестах  $n$  или  $m$  равно 2.

### Примеры

стандартный ввод	стандартный вывод
1 1	8
1 2	18
2 2	44
3 5	196

### Замечание

Все треугольники из первого примера:



## Задача 3. Порядок во всём

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася — очень порядочный мальчик, он любит порядок во всём.

У него в тетради есть столбик натуральных чисел, и он хочет изменить его так, чтобы числа шли по порядку, то есть по неубыванию. При этом Вася, естественно, ничего не хочет зачёркивать, поэтому единственное, что ему остаётся — это дописать цифры в конец некоторых чисел.

Вася хочет, чтобы после дописывания цифр последнее число в списке оказалось наименьшим возможным. Найдите это число.

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  ( $2 \leq n \leq 3 \cdot 10^5$ ) — количество чисел в тетрадке у Васи.

Следующие  $n$  строк содержат  $n$  чисел, записанных в тетрадке, по одному в каждой строке. Все числа натуральные, не превосходящие  $10^9$ .

### Формат выходных данных

Программа должна вывести наименьшее число, которое могло оказаться у Васи в конце списка.

### Система оценки

Решения, верно работающие, когда  $n \leq 5$  и при этом все числа в списке однозначные, будут оцениваться в 10 баллов.

Решения, верно работающие, когда  $n \leq 5$  и при этом все числа в списке не превосходят 999, будут оцениваться в 20 баллов.

Решения, верно работающие, когда  $n \leq 5$  без дополнительных ограничений на числа, будут оцениваться в 40 баллов.

Решения, верно работающие, когда  $n \leq 1000$  без дополнительных ограничений на числа, будут оцениваться в 60 баллов.

### Примеры

стандартный ввод	стандартный вывод
3 1 5 7	7
2 13 1	13
3 20 2 1	100

### Замечание

В первом примере числа уже упорядочены, Васе не нужно ничего дописывать.

Во втором примере Васе можно приписать ко второму числу цифру 3, тогда числа станут равны 13, а значит, будут расположены по неубыванию. При этом 13 — это минимально возможное последнее число.

В третьем примере Вася может, например, получить числа 20, 25, 100. Возможны и другие варианты, но последнее число при любом способе дописывания цифр получится не меньше 100.

## Задача 4. Тройка

Ограничение по времени: 2.5 секунд  
Ограничение по памяти: 256 мегабайт

Арсений очень любит пользоваться городским транспортом. В городе, где он живёт, существует карта «Тройка», позволяющая оплачивать проезд при помощи тарифа «Кошелёк». Есть два вида тарифа:

- «Единый» (57 рублей) — одна поездка на любом виде транспорта;
- «90 минут» (85 рублей) — не более одной поездки на метро и любое количество поездок на наземном транспорте в течение не более 90 минут с момента начала первой поездки (между началом поездки и началом первой поездки должно пройти не более 90 минут).

Так как Арсений коллекционирует карты «Тройка», у него их очень много, поэтому он может использовать неограниченное количество билетов одновременно.

У него есть планы на ближайшие  $n$  поездок. Помогите мальчику узнать, какое минимальное количество денег он должен потратить для реализации своих планов.

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  — количество поездок, которые были запланированы,  $1 \leq n \leq 10^5$ .

Следующие  $n$  строк содержат два значения, разделённые пробелом. Сначала указан вид транспорта: заглавная английская буква «В», если Арсений будет использовать наземный транспорт, или заглавная английская буква «М», если он воспользуется метро. Затем указано время начала поездки в формате ЧЧ:ММ (в виде двузначного количества часов и затем двузначного количества минут).

Поездки указаны в порядке их совершения, но они могут занимать несколько последовательных дней. Если время, записанное в какой-то строке, меньше, чем время в предыдущей строке, то данная поездка была совершена на следующий день. При этом гарантируется, что в каждый день Арсений совершит хотя бы одну поездку.

Также гарантируется, что разница времени совершения двух поездок составляет не менее 10 минут.

### Формат выходных данных

Программа должна вывести одно целое число — сколько денег потратит Арсений, если будет максимально эффективно использовать карты.

### Система оценки

Решения, правильно работающие при  $n \leq 10$ , будут набирать не менее 10 баллов.

Решения, правильно работающие, когда все поездки были совершены на метро, будут набирать не менее 15 баллов.

Решения, правильно работающие, когда все поездки были совершены на наземном транспорте, будут набирать не менее 25 баллов.

Решения, правильно работающие, когда не было совершено более двух поездок на наземном транспорте подряд, будут набирать не менее 35 баллов.

Решения, правильно работающие, когда все поездки были совершены в один день, будут набирать не менее 25 баллов.

## Примеры

стандартный ввод	стандартный вывод
3 M 10:00 B 10:20 B 11:00	85
4 B 23:59 M 00:29 M 00:59 B 01:29	142
4 B 22:00 B 23:00 B 23:50 B 00:30	142

## Замечание

В первом примере все три поездки могут быть оплачены одним тарифом «90 минут» за 85 рублей.

Во втором примере нужно одним билетом «90 минут» за 85 рублей оплатить первую (23:59), вторую (00:29) и четвёртую (01:29) поездки. Третью поездку (00:59) нельзя оплатить тем же билетом, потому что в тарифе «90 минут» может быть не более одной поездки на метро, для этой поездки придётся использовать отдельный билет за 57 рублей.

В третьем примере первую поездку (22:00) нужно оплатить отдельным билетом за 57 рублей, а следующие три поездки (23:00, 23:50, 00:30) — билетом «90 минут».

## Задача 5. Все на съезд!

Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

В 2025 году в Берляндии впервые будет проводиться трёхдневный межпланетный съезд по вопросам проведения олимпиад по информатике. Доклады съезда разбиты на 12 секций, и теперь организаторам необходимо распределить секции по дням: в каждый день будут проводиться 4 секции.

Известно, что в съезде примут участие  $n$  человек. Каждый участник съезда выбрал 3 секции, которые он хочет посетить. Но поскольку в один день секции будут проводиться одновременно, каждый участник в один день может присутствовать не более чем на одной секции. Поэтому если в один день будут идти две или три секции, выбранные каким-то участником, то он всё равно сможет посетить только одну из них. Если же выбранные секции будут проходить в разные дни, участник сможет посетить их все.

Для того чтобы съезд принёс как можно больше пользы, необходимо составить расписание съезда таким образом, чтобы суммарное число секций, посещённых всеми участниками, было как можно больше. Помогите оргкомитету составить такое расписание.

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 10\,000$ ) — количество участников съезда.

В каждой из следующих  $n$  строк даны 3 попарно различных натуральных числа, не превосходящие 12, — номера секций, которые хочет посетить один из участников.

### Формат выходных данных

Программа должна вывести 3 строки, в каждой из которых должны быть 4 числа через пробел — номера секций, проводимых в первый, второй и третий день съезда соответственно. Каждое из чисел от 1 до 12 должно встречаться в выводе ровно один раз. Если возможных оптимальных расписаний несколько, можно вывести любое из них.

### Система оценки

- Решения, правильно работающие, когда  $n = 1$ , будут оцениваться в 10 баллов.
- Решения, правильно работающие, когда  $n = 2$ , будут оцениваться в 20 баллов.
- Решения, правильно работающие, когда  $n = 3$ , будут оцениваться в 20 баллов.
- Решения, правильно работающие, когда  $n \leq 100$ , будут оцениваться в 75 баллов.

### Пример

стандартный ввод	стандартный вывод
3	1 11 6 12
5 6 1	10 5 7 8
6 7 9	9 2 3 4
1 9 7	

### Замечание

В примере из условия расписание составлено так, что второй и третий участник посетят все желаемые секции, а первый — две секции (5 и одну из секций 1, 6). Таким образом, суммарно будут посещены 8 секций. Можно показать, что этот результат улучшить нельзя.