

1. [Sort an Array - LeetCode](#)

2. Ломаная.

Задано N точек на плоскости. Указать $(N-1)$ -звенную несамопересекающуюся незамкнутую ломаную, проходящую через все эти точки.

Указание: стройте ломаную в порядке возрастания x -координаты. Если имеются две точки с одинаковой x -координатой, то расположите раньше ту точку, у которой y -координата меньше.

| Ввод | Вывод |
|------|-------|
| 4 | 0 0 |
| 0 0 | 0 1 |
| 1 1 | 1 0 |
| 1 0 | 1 1 |
| 0 1 | |

3. База данных.

В базе данных хранится N записей, вида $(Name, a_1, a_2, \dots, a_k)$ — во всех записях одинаковое число параметров. На вход задачи подаётся приоритет полей — перестановка на числах $1, \dots, k$ — записи нужно вывести в соответствии с этим приоритетом. В случае, если приоритет полей таков: 3 4 2 1, то это следует воспринимать так: надо читать как: приоритет значений из 3 колонки самый высокий, приоритет значений из колонки 4 ниже, приоритет значений из колонки 2 ещё ниже, а приоритет значений из колонки 1 самый низкий. Можно использовать стандартную сортировку.

Формат ввода:

N ($1 \leq N \leq 1000$) k ($1 \leq k \leq 10$) $p_1 p_2 \dots p_k$ (перестановка на k числах, разделитель пробел) N строк вида $Name a_1 a_2 \dots a_k$ (разделитель — пробел)

Формат вывода:

N строк с именами согласно приоритету

| Входные данные | Выходные данные |
|----------------|-----------------|
| 3 | A |
| 3 | C |
| 2 1 3 | B |
| A 1 2 3 | |
| B 2 1 3 | |
| C 3 1 2 | |

4. Современники.

Группа людей называется современниками если был такой момент, когда они могли собраться вместе. Для этого в этот момент каждому из них должно было уже исполниться 18 лет, но ещё не исполниться 80 лет. Дан список Жизни Великих Людей. Необходимо получить максимальное количество современников. В день 18летия человек уже может принимать участие в собраниях, а в день 80летия и в день смерти уже не может.

Замечание. Человек мог не дожить до 18-летия, либо умереть в день 18-летия. В этих случаях принимать участие в собраниях он не мог. Можно использовать стандартную сортировку.

Формат ввода:

Сначала на вход программы поступает число N — количество людей (меньше 10000).

Далее в N строках вводится по 6 чисел — первые три задают дату (день, месяц, год) рождения, следующие три — дату смерти (она всегда не ранее даты рождения). День (в зависимости от месяца, а в феврале — еще и года) от 1 до 28, 29, 30 или 31, месяц — от 1 до 12, год — от 1 до 2005.

Формат вывода:

Единственное число – максимальное количество современников.

| Ввод | Вывод |
|--|-------|
| 3 2 5 1980 13 11 2055 1 1 1982 1 1 2030 2 1 1920 2 1 2000 | 3 |

5. Первые k элементов длинной последовательности.

Дана очень длинная последовательность целых чисел длины n . Требуется вывести в отсортированном виде её наименьшие k элементов. Последовательность может не помещаться в память. Время работы $O(n * \log(k))$. Доп. память $O(k)$. Использовать слияние.

Формат ввода:

Сначала n , потом k , потом последовательность.

| Ввод | Вывод |
|---------------------------|---------|
| 9 4 3 7 4 5 6 1 15 4 2 | 1 2 3 4 |

6. Количество инверсий

Дана последовательность целых чисел из диапазона $(-1000000000 .. 1000000000)$. Длина последовательности не больше 1000000. Числа записаны по одному в строке. Количество чисел не указано. Пусть количество элементов n , и числа записаны в массиве $a = a[i]$: i из $[0..n-1]$. Требуется напечатать количество таких пар индексов (i, j) из $[0..n-1]$, что $(i < j$ и $a[i] > a[j])$. Указание: количество инверсий может быть больше $4 * 1000000000$ - используйте `int64_t`.

| Ввод | Вывод |
|---------|-------|
| 1 2 3 4 | 0 |
| 4 3 2 1 | 6 |
| 3 2 2 | 2 |