

Задача 1. Таблица инверсий

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Требуется реализовать функцию, которая для заданной перестановки строит соответствующую ей таблицу инверсий, и решить с её помощью тестовую задачу.

Функция должна иметь сигнатуру:

```
void permut_to_invtab (int a[], int b[], int n);
```

Здесь `a` — имя массива, содержащего перестановку, `n` — его длина, а `b` — имя массива, в который нужно записать построенную таблицу инверсий.

Формат входных данных

В первой строке входного файла записано целое число N — длина перестановки ($1 \leq N \leq 1000$).

Во второй строке через пробел записаны различные натуральные числа a_1, a_2, \dots, a_N , принимающие значения от 1 до N — перестановка.

Формат выходных данных

В выходной файл необходимо вывести через пробел N целых чисел, которые будут образовывать таблицу инверсий для заданной перестановки.

Пример

input.txt	output.txt
8 5 2 7 3 8 6 4 1	7 1 2 4 0 2 0 0

Задача 2. Сортировка выбором

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 9 секунд
Ограничение по памяти: разумное

Требуется реализовать функцию, которая сортирует заданный массив методом выбора, и решить с её помощью тестовую задачу.

Функция должна иметь сигнатуру:

```
void select_sort (int a[], int n);
```

Здесь `a` — имя массива целых чисел, а `n` — его длина.

Формат входных данных

В первой строке входного файла записано целое число N — длина последовательности ($1 \leq N \leq 10^5$).

В следующей строке через пробел записано N целых чисел. Все числа по модулю не превосходят 10^6 .

Формат выходных данных

В выходной файл необходимо вывести заданную последовательность в отсортированном по возрастанию виде. Числа выводить через пробел в одну строку.

Пример

input.txt	output.txt
5 12 5 1 -3 4	-3 1 4 5 12

Задача 3. Сортировка вставками

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 6 секунд
Ограничение по памяти: разумное

Требуется реализовать функцию, которая сортирует заданный массив методом простых вставок, и решить с её помощью тестовую задачу.

Функция должна иметь сигнатуру:

```
void ins_sort (int a[], int n);
```

Здесь `a` — имя массива целых чисел, а `n` — его длина.

Формат входных данных

В первой строке входного файла записано целое число N — длина последовательности ($1 \leq N \leq 10^5$).

В следующей строке через пробел записано N целых чисел. Все числа по модулю не превосходят 10^6 .

Формат выходных данных

В выходной файл необходимо вывести заданную последовательность в отсортированном по возрастанию виде. Числа выводить через пробел в одну строку.

Пример

input.txt	output.txt
5 12 5 1 -3 4	-3 1 4 5 12

Задача 4. Шейкер-сортировка

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	14 секунд
Ограничение по памяти:	разумное

Требуется реализовать функцию, которая упорядочивает по возрастанию заданный массив методом шейкер-сортировки, и решить с её помощью тестовую задачу.

Функция должна иметь сигнатуру:

```
void shake_sort (int a[], int n);
```

Здесь `a` — имя массива целых чисел, а `n` — его длина.

Формат входных данных

В первой строке входного файла записано целое число N — длина последовательности ($1 \leq N \leq 10^5$).

В следующей строке через пробел записано N целых чисел. Все числа по модулю не превосходят 10^6 .

Формат выходных данных

В выходной файл необходимо вывести заданную последовательность в отсортированном по возрастанию виде. Числа выводить через пробел в одну строку.

Пример

input.txt	output.txt
5 12 5 1 -3 4	-3 1 4 5 12

Задача 5. Восстановление перестановки

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Требуется реализовать функцию, которая по заданной таблице инверсий восстанавливает соответствующую ей перестановку, и решить с её помощью тестовую задачу.

Функция должна иметь сигнатуру:

```
int invtab_to_permut (int b[], int a[], int n);
```

Здесь `b` — имя массива, содержащего таблицу инверсий, `n` — его длина, `a` — имя массива, в который нужно записать восстановленную перестановку.

Функция возвращает 1, если восстановление прошло корректно, иначе она возвращает 0.

Формат входных данных

В первой строке входного файла записано целое число N — длина таблицы инверсий ($1 \leq N \leq 10^3$).

Во второй строке через пробел записаны натуральные числа a_1, a_2, \dots, a_N , принимающие значения в диапазоне от 0 до $N - 1$, образующие таблицу инверсий.

Формат выходных данных

В выходной файл необходимо вывести соответствующую заданной таблице инверсий перестановку.

Если таблица инверсий задана некорректно, то вывести слово NO.

Примеры

input.txt	output.txt
8 7 1 2 4 0 2 0 0	5 2 7 3 8 6 4 1
3 1 0 1	NO

Задача 6. Следующая по алфавиту перестановка

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Требуется реализовать функцию, которая для заданной перестановки чисел от 1 до N строит следующую за ней по алфавиту перестановку, и решить с её помощью тестовую задачу.

Функция должна иметь сигнатуру:

```
void next_permut ( int a[], int n);
```

Здесь a — имя массива, содержащего перестановку, n — его длина.

Функция должна записать следующую по алфавиту перестановку в тот же массив, не используя дополнительной памяти. Если исходная перестановка — последняя по алфавиту, то следующей для нее будет первая по алфавиту.

Формат входных данных

В первой строке входного файла записано целое число N — длина перестановки ($1 \leq N \leq 10^3$).

Во второй строке через пробел записаны различные натуральные числа a_1, a_2, \dots, a_N , принимающие значения от 1 до N .

Формат выходных данных

В выходной файл необходимо вывести через пробел N заданных чисел, которые будут образовывать следующую по алфавиту перестановку для заданной.

Пример

<code>input.txt</code>	<code>output.txt</code>
8 5 2 7 3 8 6 4 1	5 2 7 4 1 3 6 8

Задача 7. Сортировка по ключу

Источник: основная*
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

В первой строке записано целое число N — количество записей ($1 \leq N \leq 10\,000$). В остальных N строках содержатся записи, по одной в строке.

Для каждой записи указаны ключ и значение через пробел. Ключ — это целое число в диапазоне от 0 до 10^9 включительно, а значение — это строка от одного до семи символов включительно, состоящая только из маленьких букв латинского алфавита.

Требуется вывести ровно те же самые N записей, но в другом порядке. Записи должны быть упорядочены по возрастанию ключа. Если у нескольких записей ключ равный, то нужно упорядочить их в том порядке, в котором они встречаются по входному файлу.

Пример

input.txt	output.txt
7	1 a
3 qwerty	2 hello
3 string	3 qwerty
6 good	3 string
1 a	3 ab
3 ab	5 world
2 hello	6 good
5 world	

Пояснение к примеру

В примере 7 записей с ключами 1, 2, 3, 5 и 6 — именно в таком порядке записи и выведены в выходном файле. Обратите внимание, что есть три записи с ключом 3: `qwerty`, `string`, `ab`. Они выведены ровно в том порядке, в котором они идут во входном файле.

Задача 8. Сортировка со списками

Источник:	повышенной сложности*
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

В первой строке записано целое число N — количество записей ($1 \leq N \leq 2 \cdot 10^5$). В остальных N строках содержатся записи, по одной в строке.

Для каждой записи указаны ключ и значение через пробел. Ключ — это целое число в диапазоне от 0 до 10^6 включительно, а значение — это строка от одного до семи символов включительно, состоящая только из маленьких букв латинского алфавита.

Требуется вывести ровно те же самые N записей, но в другом порядке. Записи должны быть упорядочены по возрастанию ключа. Если у нескольких записей ключ равный, то нужно упорядочить их в том порядке, в котором они встречаются по входном файле.

Важно: Решать задачу **нужно** следующим образом (другие решения засчитываться **не** будут). Нужно завести 10^6 связанных списков, и в каждый k -ый список складывать все записи с ключом, равным k . Тогда после раскидывания записей по спискам достаточно будет пробежаться по спискам в порядке увеличения k и распечатать их.

Пример

<code>input.txt</code>	<code>output.txt</code>
7	1 a
3 qwerty	2 hello
3 string	3 qwerty
6 good	3 string
1 a	3 ab
3 ab	5 world
2 hello	6 good
5 world	

Пояснение к примеру

В примере 7 записей с ключами 1, 2, 3, 5 и 6 — именно в таком порядке записи и выведены в выходном файле. Обратите внимание, что есть три записи с ключом 3: `qwerty`, `string`, `ab`. Они выведены ровно в том порядке, в котором они идут во входном файле.

Задача 9. Функция

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Пусть P — множество целых чисел от 1 до N . Задано отношение R на множестве $P \times P$. Отношение задаётся списком принадлежащих ему элементов: множеством пар $(x, y) \in P \times P$.

Для заданного отношения R требуется определить:

1. Является ли R функцией: $\forall x \in P: (x, u) \in R$ и $(x, v) \in R \Rightarrow u = v$ — отсутствует многозначность.
2. Является ли R всюду определённой функцией: R является функцией и $\forall x \in N \exists y \mid (x, y) \in R$ — значение определено на всём множестве P .
3. Является ли R инъекцией: $(x, u) \in R$ и $(y, u) \in R \Rightarrow x = y$.
4. Является ли R сюръекцией: $\forall u \in N \exists x \mid (x, u) \in R$.
5. Является ли R биекцией: отношение R и инъективно, и сюръективно.

Формат входных данных

Первая строка входного файла содержит два целых числа N и M , записанных через пробел — размер множества и количество пар ($1 \leq N \leq 300, 1 \leq M \leq N^2$).

В следующих M строках записано по два целых числа x и y — элементы отношения R ($1 \leq x, y \leq N$). Гарантируется, что все пары различны.

Формат выходных данных

Если отношение R не удовлетворяет ни одному из описанных свойств, то в выходной файл нужно вывести число 0.

В противном случае необходимо вывести через пробел номера свойств, которыми обладает отношение R , в порядке увеличения значений.

Примеры

input.txt	output.txt
3 5 1 2 3 3 1 3 1 1 2 1	0
5 2 1 3 2 3	1
5 2 1 3 2 5	1 3
3 3 1 1 2 3 3 2	1 2 3 4 5

Задача 10. Бинарный поиск+

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда*
Ограничение по памяти:	разумное

Требуется реализовать функцию, которая находит методом бинарного поиска элемент с заданным значением в отсортированном массиве, и решить с её помощью тестовую задачу.

Функция должна иметь сигнатуру:

```
int binsearch (int a[], int l, int r, int x);
```

Здесь a — имя массива целых чисел, l и r — индексы элементов, соответствующие левой и правой границам поиска, x — значение искомого элемента.

Функция `binsearch` возвращает индекс элемента в массиве, равного x . Если таких элементов несколько, то выдается самый правый индекс. Если элемент в массиве отсутствует, то функция выдает -1 .

Формат входных данных

В первой строке записано одно целое число N — размер отсортированного массива ($1 \leq N \leq 10^5$).

Далее записаны элементы массива A_i (N целых чисел, $|A_i| \leq 10^9$).

Затем записано целое число Q — количество запросов, которые нужно обработать ($1 \leq Q \leq 10^5$).

В остальных Q строках записаны целые числа X_j , определяющие запросы на поиск.

Формат выходных данных

Каждый запрос нужно обрабатывать следующим образом. Сначала нужно прибавить к записанному в файле числу X_j ответ на предыдущий запрос R_{j-1} , получив $Y_j = X_j + R_{j-1}$. Затем нужно найти в массиве A элемент, равный Y_j ; его индекс будет ответом R_j для этого запроса. Если таких элементов много, то в качестве ответа R_j следует выбрать самый большой индекс. Если таких элементов нет, то ответ R_j равен -1 .

Элементы массива нумеруются индексами от 0 до $N - 1$. Для первого запроса предыдущего ответа нет, так что полагаем $Y_0 = X_0$.

Пример

input.txt	output.txt
10	1
1 1 3 4 4 7 8 10 10 12	2
10	-1
1	2
2	5
3	-1
4	-1
5	5
6	-1
7	-1
8	
9	
10	

Пояснение к примеру

Первый запрос: нужно найти значение $X_0 = Y_0 = 1$. Таких элементов два и они имеют индексы 0 и 1. В данной задаче нужно всегда выбирать максимальных индекс, если выбор есть, поэтому ответ A_0 равен 1.

Для следующего запроса задано число $X_1 = 2$. Прибавляем к нему предыдущий ответ $A_0 = 1$, и получаем число $Y_1 = 3$, которое нужно искать. Такой элемент есть в массиве под индексом 2, так что выводим ответ $A_1 = 2$.

Для следующего запроса указано $X_2 = 3$. Прибавляем предыдущий ответ $A_1 = 2$, и получаем, что нужно искать $Y_2 = 5$. Такого числа нет, так что выводим ответ $A_2 = -1$.

Теперь рассмотрим запрос $X_3 = 4$. Сперва прибавляем предыдущий ответ $A_2 = -1$, получаем число $Y_3 = 3$, которое нужно искать. Значит ответ $A_3 = 2$. И так далее...