

Задача 1. Сравнение элементов

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

Дан массив из N чисел: b_1, b_2, \dots, b_N .

Требуется для каждого элемента массива b_i посчитать количество элементов b_j , стоящих правее ($i < j$), но меньших него ($b_i > b_j$).

Формат входных данных

В первой строке записано целое число N ($1 \leq N \leq 100$).

Во второй строке через пробел записано N целых положительных чисел — значения элементов массива. Гарантируется, что значения элементов не превосходят 100.

Формат выходных данных

Выведите N целых чисел через пробел: по порядку для каждого элемента входного массива выведите количество элементов, удовлетворяющих описанному выше условию.

Примеры

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

Пояснение к примеру

В первом примере:

- Правее числа 4 стоят числа 3, 5, 1 и 2, **три** из которых имеют значение меньше 4.
- Правее числа 3 стоят числа 5, 1 и 2, **два** из которых имеют значение меньше 3.
- Правее числа 5 стоят числа 1 и 2, **два** из которых имеют значение меньше 5.
- Для чисел 1 и 2 справа нет чисел, имеющих значение меньше.

Задача 2. Суммы k -ых

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

Дан массив из N чисел a_1, a_2, \dots, a_N .

Требуется вывести N чисел t_k ($k = 1, \dots, N$), где t_k — сумма элементов массива с шагом k :

$$t_k = \sum_{j=1}^{N/k} a_{j \cdot k}$$

Т.е. t_1 равняется сумме всех элементов массива, t_2 равняется сумме каждого второго элемента массива, t_3 равняется сумме каждого третьего элемента и т.д.

Формат входных данных

В первой строке записано целое число N ($1 \leq N \leq 10^5$).

Во второй строке через пробел записано N целых чисел a_i ($1 \leq a_i \leq 10^4$).

Формат выходных данных

Выведите N строк: в k -й строке выведите значение t_k .

Пример

input.txt	output.txt
6	24
4 3 5 1 2 9	13
	14
	1
	2
	9

Пояснение к примеру

$$t_1 = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 = 4 + 3 + 5 + 1 + 2 + 9 = 24$$

$$t_2 = a_2 + a_4 + a_6 = 3 + 1 + 9 = 13$$

$$t_3 = a_3 + a_6 = 5 + 9 = 14$$

$$t_4 = a_4 = 1$$

$$t_5 = a_5 = 2$$

$$t_6 = a_6 = 9$$

Задача 3. Гистограмма

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

Дан массив из N чисел a_0, a_1, \dots, a_{N-1} .

Постройте гистограмму значений элементов массива: для каждого значения подсчитайте сколько раз оно встречается в массиве.

Формат входных данных

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

Во второй строке через пробел записано N целых чисел a_i ($1 \leq a_i \leq 10^4$).

Формат выходных данных

Для каждого значения, которое встречается в массиве хотя бы раз, в выходной файл требуется вывести сколько раз оно присутствует среди элементов массива в формате: “значение: количество”.

Значения требуется выводить в порядке возрастания.

Для вывода двух целых чисел, разделённых двоеточием с пробелом, удобно использовать функцию `printf` со следующей форматной строкой:

```
printf("%d: %d", value, count);
```

Пример

input.txt	output.txt
10	1: 2
3 4 5 10 3 4 10 1 1 3	3: 3
	4: 2
	5: 1
	10: 2

Задача 4. Различные числа

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

Задана последовательность целых чисел a_1, a_2, \dots, a_n . Сосчитать, сколько различных чисел в этой последовательности. Например, в массиве из четырех чисел 5,7,5,5 различных чисел два.

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 100000$).

Во второй строке через пробел записаны целые числа a_1, a_2, \dots, a_n , каждое из которых по модулю не превосходит 10^3 .

Формат выходных данных

В выходной файл необходимо вывести целое число – количество различных чисел.

Примеры

input.txt	output.txt
10 1 2 3 5 4 1 2 2 1 4	5

Задача 5. Делимость на 15

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

На вход подается число. Нужно определить делится ли оно на 15.

Формат входных данных

Входной файл из последовательности цифр, за которыми следует символ «точка». Длина числа не превышает 500.

Формат выходных данных

Если введенное число делится нацело на 15, то в выходной файл необходимо вывести слово YES. В противном случае необходимо вывести слово NO.

Примеры

<code>input.txt</code>	<code>output.txt</code>
12345.	YES
67.	NO

Задача 6. Игра в алфавит

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

Два товарища играют в слова. Игроки по очереди приводят примеры слов из некоторой предметной области. Начинает Игрок 1. На каждом ходу игрок должен привести пример, который содержит указанную букву алфавита, которая изменяется на следующем шаге. В частности, Игрок 1 должен сначала привести пример слова, содержащего букву 'а'. Игрок 2 должен затем привести пример слова, содержащего букву 'b'. Игрок 1 должен на следующем шаге привести пример слова, содержащего 'с', и так далее. Если игрок не может назвать соответствующее слово, то он проигрывает, а другой игрок, соответственно, выигрывает.

Для заданной последовательности слов, которые по очереди произносили игроки, вам нужно написать программу, которая бы определяла, кто победил. Проигравшим считается тот игрок, который первым не смог привести соответствующий пример слова.

Формат входных данных

Во входном файле записана непустая строка, содержащая последовательность слов, разделенных одним пробелом. Все слова содержат маленькие латинские буквы. Количество слов не превышает 26. Длина одного слова может изменяться от 1 до 20.

Формат выходных данных

В выходной файл необходимо вывести одну из трёх возможных комбинаций слов. Если выигрывает первый игрок, то в выходной файл нужно вывести `PLAYER 1`. Если же выигрывает второй игрок, нужно вывести `PLAYER 2`. Если никто не выигрывает, то вывести `NO WINNER`.

Примеры

<code>input.txt</code>	<code>output.txt</code>
<code>alpha beta gamma delta epsilon</code>	<code>PLAYER 2</code>
<code>january february march april may</code>	<code>PLAYER 1</code>
<code>cab cab cab ford chevy ford dodge honda</code>	<code>NO WINNER</code>

Задача 7. Количество слов

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

В единственной строке файла записан набор слов, разделённых символами точки. Каждое слово состоит из букв латинского алфавита. Между словами находится один или несколько символов точки. До первого слова и после последнего точки могут быть, а могут не быть. В строке может быть записано всего одно слово, а может и вовсе не быть слов. Длина заданной строки находится в диапазоне от 1 до 10 000.

Требуется вывести одно целое число — количество слов в строке.

Пример

<code>input.txt</code>	<code>output.txt</code>
<code>..ko..Privet.kreved....ko...</code>	4

Пояснение к примеру

В примере записано четыре слова: `ko`, `Privet`, `kreved` и `ko`.

Комментарий

В данной задаче нужно читать символы из входного файла по одному, сохраняя их в переменную типа `char`. Примерно так:

```
char symbol;  
scanf("%c", &symbol);
```

Гарантируется, что после строки имеется символ перевода строки `'\n'`.

Задача 8. Сумма чисел

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

В заданном тексте нужно посчитать сумму входящих в него десятичных целых чисел.

Формат входных данных

Во входном файле записана непустая последовательность символов, среди которых могут встречаться цифры. Эта последовательность заканчивается символом «точка». Длина последовательности не превосходит 10^4 . Числа, которые в ней встречаются, имеют не более 5 знаков.

Формат выходных данных

В выходной файл необходимо вывести целое число – сумму чисел, которые встречаются в заданной последовательности.

Пример

input.txt	output.txt
This is example: three (3) + 15 is equal to 18.	36

Задача 9. Разность множеств

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

Дано два множества символов $S1$ и $S2$. Вам нужно вычислить их разность ($S1 - S2$) – новое множество, в котором присутствуют только те элементы $S1$, которых нет в $S2$.

Формат входных данных

В единственной строке входного файла через пробел записано две непустые последовательности латинских букв в верхнем регистре. Эти последовательности задают два множества – $S1$ и $S2$, соответственно. Каждый элемент множества встречается в последовательности не более одного раза. Количество элементов в каждом множестве не более 26.

Формат выходных данных

В выходной файл необходимо вывести последовательность символов, которые будут представлять разность заданных множеств. Элементы искомого множества выдавать в возрастающем порядке. Если результирующее множество – пусто, то вывести `Empty Set`.

Примеры

<code>input.txt</code>	<code>output.txt</code>
AEIOU AEIOUY	Empty Set
PQRSTUV UVWXY	PQRST
TEXAS COMPUTER	ASX

Задача 10. Слова

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

Перечислить все слова из заданной строки, не содержащие букв, из которых состоит первое слово.

Формат входных данных

Во входном файле записана строка, состоящая из слов, разделенных одним пробелом. Каждое слово состоит из маленьких латинских букв.

Формат выходных данных

В выходной файл необходимо вывести все слова, по одному на строку, обладающие заданным свойством, в порядке их встречаемости во входном файле.

Пример

<code>input.txt</code>	<code>output.txt</code>
to be or not to be that is the question	be be is

Задача 11. Построение списка добавлением в голову

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

По заданной последовательности целых чисел построить односвязный динамический список. Каждое новое число добавлять в начало списка. Затем пройти по построенному списку и посчитать количество отрицательных чисел, входящих в список, кратных 7. После этого память освободить.

Формат входных данных

Входной файл содержит заданную последовательность целых чисел. Числа в файле записаны через пробел. Их величина по модулю не превосходит 1000. Количество чисел может изменяться от 1 до 1000.

Формат выходных данных

В выходной файл нужно вывести одно целое число — количество отрицательных чисел, кратных 7.

Пример

<code>input.txt</code>	<code>output.txt</code>
10 -1 14 8 -21 -35 35 16	2

Задача 12. Построение списка добавлением в хвост

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

По заданной последовательности целых чисел построить односвязный динамический список. Каждое новое число добавлять в конец списка. Затем пройти по построенному списку и посчитать среднее арифметическое чисел, входящих в список. После этого память освободить.

Формат входных данных

Входной файл содержит заданную последовательность целых чисел. Числа в файле записаны через пробел. Их величина по модулю не превосходит 1000. Количество чисел может изменяться от 1 до 1000.

Формат выходных данных

В выходной файл нужно вывести одно целое число — среднее арифметическое элементов списка.

Пример

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

Задача 13. Удаление предшественников

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

По заданной последовательности целых чисел построить односвязный список. Каждое новое число добавлять в начало списка. Затем пройти по построенному списку и удалить элемент, предшествующий элементу, содержащему заданное число, для всех таких вхождений. После этого распечатать полученный список, память освободить.

Формат входных данных

Первая строка входного файла содержит заданное число, которое нужно удалить.

В следующей строке записана последовательность целых чисел. Числа в файле записаны через пробел. Их величина по модулю не превосходит 1000. Количество чисел может изменяться от 1 до 1000.

Формат выходных данных

В выходной файл нужно вывести последовательность элементов, оставшихся в списке после удаления.

Пример

input.txt	output.txt
4 1 5 4 6 4 3	4 4 5 1

Задача 14. Удаление повторов

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

Из списка, элементами которого являются целые числа, убрать повторения (т.е. подряд не должны идти одинаковые элементы.)

Формат входных данных

Во входном файле через пробел записаны целые числа. Из этих чисел нужно построить односвязный список в порядке их встречаемости во входном файле. Количество чисел от 1 до 10^5 .

Формат выходных данных

Из полученного списка нужно удалить все повторяющиеся элементы, а затем в выходной файл выдать через пробел оставшиеся числа. Память освободить.

Пример

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

Задача 15. Построение упорядоченного списка

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

По заданной последовательности целых чисел построить односвязный динамический список. Каждое новое число добавлять в список так, чтобы он оставался упорядоченным по возрастанию. Если такое число в списке уже есть, то его не добавлять. Затем пройти по построенному списку от начала до конца и распечатать его элементы. После этого память освободить.

Формат входных данных

Входной файл содержит заданную последовательность целых чисел. Числа в файле записаны через пробел. Их величина по модулю не превосходит 1000. Количество чисел может изменяться от 1 до 1000.

Формат выходных данных

В выходной файл нужно вывести упорядоченную последовательность заданных чисел без повторений. Числа выводить через пробел в одну строку.

Пример

<code>input.txt</code>	<code>output.txt</code>
10 -1 14 8 -21 -3 35 16 -3 10	-21 -3 -1 8 10 14 16 35

Задача 16. Удаление последнего

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

По заданной последовательности целых чисел построить односвязный список. Каждое новое число добавлять в конец списка. Затем пройти по построенному списку и удалить последнее вхождение заданного числа. После этого распечатать полученный список, память освободить.

Формат входных данных

Первая строка входного файла содержит заданное число, которое нужно удалить.

В следующей строке записана последовательность целых чисел. Числа в файле записаны через пробел. Их величина по модулю не превосходит 1000. Количество чисел может изменяться от 1 до 1000.

Формат выходных данных

В выходной файл нужно вывести последовательность элементов, оставшихся в списке после удаления.

Пример

input.txt	output.txt
4 1 5 4 6 4 3	1 5 4 6 3

Задача 17. Удаление первого

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

По заданной последовательности целых чисел построить односвязный список. Каждое новое число добавлять в начало списка. Затем пройти по построенному списку и удалить первое вхождение заданного числа. После этого распечатать полученный список, память освободить.

Формат входных данных

Первая строка входного файла содержит заданное число, которое нужно удалить.

В следующей строке записана последовательность целых чисел. Числа в файле записаны через пробел. Их величина по модулю не превосходит 1000. Количество чисел может изменяться от 1 до 1000.

Формат выходных данных

В выходной файл нужно вывести последовательность элементов, оставшихся в списке после удаления.

Пример

input.txt	output.txt
4 1 5 4 6 4 3	3 6 4 5 1

Задача 18. Удаление всех

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

По заданной последовательности целых чисел построить односвязный список. Каждое новое число добавлять в конец списка. Затем пройти по построенному списку и удалить все вхождения заданного числа. После этого распечатать полученный список, память освободить.

Формат входных данных

Первая строка входного файла содержит заданное число, которое нужно удалить.

В следующей строке записана последовательность целых чисел. Числа в файле записаны через пробел. Их величина по модулю не превосходит 1000. Количество чисел может изменяться от 1 до 1000.

Формат выходных данных

В выходной файл нужно вывести последовательность элементов, оставшихся в списке после удаления.

Пример

input.txt	output.txt
4 1 5 4 6 4 3	1 5 6 3

Задача 19. Обходы дерева

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

По заданной последовательности целых чисел построить дерево двоичного поиска и обойти его в прямом и обратном порядках. Повторяющиеся числа в дерево не вставлять.

Формат входных данных

Во входном файле через пробел записаны целые числа. Количество чисел не превосходит 1000.

Формат выходных данных

В первую строку выходного файла нужно вывести значения, содержащиеся в построенном дереве в прямом порядке обхода, а во вторую — в обратном. Числа выводить через пробел.

Пример

<code>input.txt</code>	<code>output.txt</code>
5 1 10 -3 12 1 9 4	5 1 -3 4 10 9 12 -3 4 1 9 12 10 5

Задача 20. Высота дерева

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

По заданной последовательности целых чисел построить дерево двоичного поиска. Повторяющиеся числа в дерево не вставлять. Необходимо посчитать высоту получившегося дерева.

Формат входных данных

Во входном файле через пробел записаны целые числа. Количество чисел не превосходит 1000.

Формат выходных данных

В выходной файл нужно вывести одно целое число — высоту получившегося дерева.

Пример

<code>input.txt</code>	<code>output.txt</code>
5 1 10 -3 12 1 9 4	2

Задача 21. Количество листьев в дереве

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

По заданной последовательности целых чисел построить дерево двоичного поиска. Повторяющиеся числа в дерево не вставлять. Необходимо посчитать количество вершин, не имеющих сыновей, т.е. листьев дерева.

Формат входных данных

Во входном файле через пробел записаны целые числа. Количество чисел не превосходит 1000.

Формат выходных данных

В выходной файл нужно вывести одно целое число — количество листьев в заданном дереве.

Пример

<code>input.txt</code>	<code>output.txt</code>
5 1 10 -3 12 1 9 4	4

Задача 22. Количество вершин в дереве на заданном уровне

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

По заданной последовательности целых чисел построить дерево двоичного поиска. Повторяющиеся числа в дерево не вставлять. Необходимо посчитать количество вершин, расположенных на заданном уровне получившегося дерева.

Формат входных данных

В первой строке входного файла указан уровень дерева, на котором нужно посчитать количество вершин — это неотрицательное целое число, не превосходящее 1000.

В следующей строке через пробел записаны целые числа. Количество чисел не превосходит 1000.

Формат выходных данных

В выходной файл нужно вывести одно целое число — количество вершин, расположенных на заданном уровне получившегося дерева.

Пример

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

Задача 23. Дерево двоичного поиска слов

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

По заданным словам построить дерево двоичного поиска и обойти его в инфиксном порядке. Повторяющиеся слова в дерево не вставлять.

Формат входных данных

Входной файл содержит строки, в каждой из которых записано по одному слову. Длина каждого слова не превосходит 100 символов. Количество слов не превосходит 1000. Пустых строк нет.

В следующей строке через пробел записаны целые числа. Количество чисел не превосходит 1000.

Формат выходных данных

В выходной файл нужно выдать эти слова, упорядоченные в лексикографическом порядке, по одному на строке.

Пример

<code>input.txt</code>	<code>output.txt</code>
orange mallon apple grapes plum banana	apple banana grapes mallon orange plum