

Содержание

Must have	2
Задача 2А. Число Фибоначчи [0.2, 256]	2
Задача 2В. Сортировка вставками [0.5, 256]	3
Задача 2С. Сортировка пар [2.5, 256]	4
Задачи здорового человека	5
Задача 2D. Про спрайт [0.5, 256]	5
Задача 2Е. Собственно Куча [1, 256]	6
Задача 2F. Глубина быстрой сортировки [0.5, 256]	7
Задача 2G. Куча исправлений [0.5, 256]	8
Для искателей острых ощущений	9
Задача 2H. Маленький холодильник [2, 256]	9
Задача 2I. Менеджер памяти [1, 256]	10

У вас не получается читать/выводить данные?

Воспользуйтесь примерами (c++) (python).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Must have

Задача 2А. Число Фибоначчи [0.2, 256]

Числа Фибоначчи $F_0, F_1, F_2, \dots, F_n$ определяются следующим образом: $F_0 = F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите число Фибоначчи F_n .

Формат входных данных

В первой строке входного файла задано единственное число n ($0 \leq n \leq 45$).

Формат выходных данных

Выведите число F_n в первой строке выходного файла.

Примеры

stdin	stdout
1	1
2	2
3	3
4	5
5	8
6	13

Подсказка по решению

Просто циклы `for`.

Задача 2В. Сортировка вставками [0.5, 256]

Вам дан массив a из n целых чисел от 1 до 100. Посчитайте, сколько сравнений элементов массива сделает сортировка вставками при сортировке данного массива?

Формат входных данных

На первой строке длина массива n ($1 \leq n \leq 100$).

На второй строке массив a : n целых чисел от 1 до 100.

Формат выходных данных

Выведите одно число – количество сравнений по ходу сортировки вставками.

Примеры

stdin	stdout
5 2 2 1 1 2	7
1 3	0
8 8 8 7 1 1 2 8 2	20

Замечание

Пояснение к первому примеру:

Напомним, сортировка вставками для всех элементов $i=1..n$, ищет линейным поиском позицию, куда вставить элемент « i ».

Вставляем 2 в [], получаем [2], делаем 0 сравнений.

Вставляем 2 в [2], получаем [2,2], делаем 1 сравнение.

Вставляем 1 в [2,2], получаем [1,2,2], делаем 2 сравнения.

Вставляем 1 в [1,2,2], получаем [1,1,2,2], делаем 3 сравнения.

Вставляем 2 в [1,1,2,2], получаем [1,1,2,2,2], делаем 1 сравнение.

Жирным выделены элементы, с которыми произошло сравнение.

Подсказка по решению

Напишите сортировку вставками.

Задача 2С. Сортировка пар [2.5, 256]

Дан массив пар целых чисел a . Отсортируйте элементы массива по первому числу в паре. При равенстве первого числа, сортируйте по второму числу. При равенстве обоих элементов в паре, сортируйте по номеру пары во входных данных. В качестве ответа выведите номера пар в порядке после сортировки массива (пары нумеруются от 1 до n).

Формат входных данных

На первой строке длина массива n ($1 \leq n \leq 10^5$).

На следующих n строках элементы массива a — пары целых чисел от 1 до 10^9 .

Формат выходных данных

В одной строке выведите номера элементов массива a после его сортировки.

Примеры

stdin	stdout
9	2 6 7 3 4 9 1 5 8
3 2	
1 1	
2 1	
2 2	
3 2	
1 2	
1 2	
3 2	
2 3	

Замечание

Пояснение к первому примеру:

Давайте допишем к каждой паре её номер, чтобы уметь отличать одинаковые пары.

Сортировка массива по правилам из условия будет выглядеть так, как показано ниже.

Оригинальный массив: [(3 2 1) (1 1 2) (2 1 3) (2 2 4) (3 2 5) (1 2 6) (1 2 7) (3 2 8) (2 3 9)]

Отсортированный массив: [(1 1 2) (1 2 6) (1 2 7) (2 1 3) (2 2 4) (2 3 9) (3 2 1) (3 2 5) (3 2 8)]

Подсказка по решению

Задача о том, чтобы научиться пользоваться стандартной сортировкой.

C++: `std::sort`

python: `a.sort()`, `sorted(a)`

Задачи здорового человека

Задача 2D. Про спрайт [0.5, 256]

8б класс решил на слет взять много Спрайта. Для этого они собрались сконструировать переносной холодильник $a \times b \times c$, который будет вмещать ровно n кубических банок Спрайта размером $1 \times 1 \times 1$. Чтобы лимонад доехал как можно более холодным, они хотят минимизировать теплопотери; то есть минимизировать площадь поверхности.

Например, если емкость холодильника должна равняться 12, то возможны следующие варианты:

3 2 2 \rightarrow 32

4 3 1 \rightarrow 38

6 2 1 \rightarrow 40

12 1 1 \rightarrow 50

В этом примере оптимальным является холодильник 322.

Помогите 8б найти оптимальный холодильник в общем случае.

Формат входных данных

Число n ($1 \leq n \leq 10^6$)

Формат выходных данных

Три числа a, b, c ($1 \leq n \leq 10^6$) — размеры наилучшего холодильника.

Если оптимальных ответов несколько, выведите минимальную тройку чисел. Тройки чисел сравниваются следующим образом: сперва первое число, при равенстве второе, при равенстве третье.

Пример

stdin	stdout
12	2 2 3
13	1 1 13
1000000	100 100 100

Подсказка по решению

Просто циклы for.

Задача 2E. Собственно Куча [1, 256]

Вам даны n запросов вида
«добавить целое число от 1 до 10^9 в множество» и
«извлечь из множества минимальное по значению число».

Формат входных данных

На первой строке n ($1 \leq n \leq 200\,000$).
Далее n запросов по одному строке, подробности в примерах.
Гарантируется, что не будет извлечений из пустой кучи.

Формат выходных данных

При каждом извлечении вывести извлекаемый элемент.

Пример

stdin	stdout
6 + 1 + 2 + 1 - - -	1 1 2
5 + 4 - + 3 + 2 -	4 2

Подсказка по решению

Задача о том, чтобы научиться пользоваться стандартной библиотекой.

C++: `set`, `priority_queue`

python: `heapq`, `heapq.heappush`, `heapq.heappop`

Задача 2F. Глубина быстрой сортировки [0.5, 256]

По сути вам нужно написать свой *quick-sort*.

Дан массив целых чисел, вам предлагается написать для него быструю сортировку, которая, на каждом уровне рекурсии делает разделение по элементу, который лежит по середине массива (если длина массива чётная, нужно взять левый из двух элементов).

При разделении массив делится на три части A — элементы меньше x , B — элементы равные x , C — элементы больше x . Важно, чтобы элементы в A и C шли в том же порядке, в котором они шли в p .

Далее происходят рекурсивные вызовы от A и C . Рекурсия останавливается, если ей передали массив длины не более 1. Ваша задача — реализовать ровно описанное и запомнить максимальную глубину, на которую спускалась рекурсия.

Формат входных данных

В первой строке целое число n ($1 \leq n \leq 100$) — длина массива.

Формат выходных данных

Выведите одно число — глубину рекурсии вариции быстрой сортировки, описанной в условии.

Примеры

stdin	stdout
6 1 3 2 4 5 6	3
6 1 1 2 1 1 1	2

Замечание

Пояснение к примерам:

Глубина 0. $[1, 3, 2, 4, 5, 6]$: делимся по 2, вызываемся от $[1]$, $[3, 4, 5, 6]$.

Глубина 1. $[3, 4, 5, 6]$: делимся по 4, вызываемся от $[3]$, $[5, 6]$.

Глубина 2. $[5, 6]$ делимся по 5, вызываемся от $[], [6]$.

Глубина 3. $[6]$. Выходим из рекурсии.

Итоговая максимальная глубина 3.

Глубина 0. $[1, 1, 2, 1, 1, 1]$: делимся по 2, вызываемся от $[1, 1, 1, 1, 1]$ и $[], [2]$.

Глубина 1. $[1, 1, 1, 1, 1]$: делимся по 1, вызываемся от $[], [1]$ и $[], [1]$.

Глубина 2. $[], [1]$. Выходим из рекурсии.

Итоговая максимальная глубина 2.

Задача 2G. Куча исправлений [0.5, 256]

Жил был массив $p = [1, 2, \dots, n]$, где $n \leq 1000$. Массив испортили в $k \leq 10$ местах, в этих местах исходные числа заменили на числа от 0 до $n+1$.

Массив a называется кучей, если в a_1 хранится минимум, и $\forall i a_i \leq a_{2i}$ и $a_i \leq a_{2i+1}$.

Вам нужно за не более чем 90 swap-ов элементов сделать из массива корректную кучу.

Формат входных данных

На первой строке числа n ($1 \leq n \leq 1000$).

На следующей строке испорченная версия массива p .

Формат выходных данных

На первой строке выведите число m swap-ов, которые вы хотите совершить.

Следующие m строк должны содержать пары $i j$ – номера элементов, которые вы хотите поменять местами.

Примеры

stdin	stdout
6 1 2 3 0 5 6	2 4 2 2 1
8 1 9 3 4 5 6 7 8	2 4 2 8 4
6 5 2 3 4 5 0	4 2 1 4 2 6 3 3 1

Замечание

Пояснение к первому примеру:

Проталкиваем 0 вверх: swap(4,2), swap(2,1).

Пояснение ко второму примеру:

Проталкиваем 9 вниз: swap(2,4), swap(4,8).

Пояснение к третьему примеру:

Нужно 5 из корня протолкнуть вниз: swap(1,2), swap(2,4)

и 0 из конца массива протолкнуть вверх: swap(6,3), swap(3,1).

Подсказка по решению

Напишите свою кучу.

Для искателей острых ощущений

Задача 2Н. Маленький холодильник [2, 256]

Дано целое число n ($1 \leq n \leq 10^{12}$). Нужно найти натуральные a, b, c : $abc = n$ и при этом $2(ab + bc + ca)$ минимально. Т.е. при фиксированном объеме минимизировать площадь поверхности.

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

На первой строке четыре целые числа — $2(ab + bc + ca)$ и a, b, c .

Примеры

stdin	stdout
120	148 4 6 5

Подсказка по решению

Просто циклы `for` и пара `if`-ов.

Задача 21. Менеджер памяти [1, 256]

Пете поручили написать менеджер памяти для новой стандартной библиотеки языка C++. В распоряжении у менеджера находится массив из N последовательных ячеек памяти, пронумерованных от 1 до N . Задача менеджера – обрабатывать запросы приложений на выделение и освобождение памяти. Запрос на выделение памяти имеет один параметр K . Такой запрос означает, что приложение просит выделить ему K последовательных ячеек памяти. Если в распоряжении менеджера есть хотя бы один свободный блок из K последовательных ячеек, то он обязан в ответ на запрос выделить такой блок. При этом непосредственно перед самой первой ячейкой памяти выделяемого блока не должно располагаться свободной ячейки памяти. После этого выделенные ячейки становятся занятыми и не могут быть использованы для выделения памяти, пока не будут освобождены. Если блока из K последовательных свободных ячеек нет, то запрос отклоняется. Запрос на освобождение памяти имеет один параметр T . Такой запрос означает, что менеджер должен освободить память, выделенную ранее при обработке запроса с порядковым номером T . Запросы нумеруются, начиная с единицы. Гарантируется, что запрос с номером T – запрос на выделение, причем к нему еще не применялось освобождение памяти. Освобожденные ячейки могут снова быть использованы для выделения памяти. Если запрос с номером T был отклонен, то текущий запрос на освобождение памяти игнорируется. Требуется написать менеджер памяти, удовлетворяющий приведенным критериям.

Формат входных данных

Первая строка входного файла содержит числа N и M – количество ячеек памяти и количество запросов соответственно ($1 \leq N \leq 2^{31} - 1$; $1 \leq M \leq 10^5$). Каждая из следующих M строк содержит по одному числу: $(i+1)$ -я строка входного файла ($1 \leq i \leq M$) содержит либо положительное число K , если i -й запрос – запрос на выделение с параметром K ($1 \leq K \leq N$), либо отрицательное число $-T$, если i -й запрос – запрос на освобождение с параметром T ($1 \leq T < i$).

Формат выходных данных

Для каждого запроса на выделение памяти выведите в выходной файл результат обработки этого запроса: для успешных запросов выведите номер первой ячейки памяти в выделенном блоке, для отклоненных запросов выведите число -1 . Результаты нужно выводить в порядке следования запросов во входном файле.

Примеры

stdin	stdout
6 8	1
2	3
3	-1
-1	-1
3	1
3	-1
-5	
2	
2	