

Задача А. Двоичный поиск

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Реализуйте алгоритм бинарного поиска. Вам нужно ответить на несколько вопросов "Присутствует ли элемент x " в отсортированном массиве.

Формат входных данных

В первой строке содержатся числа n и k ($1 \leq n, k \leq 10^5$).

Во второй строке задаются n элементов первого массива, отсортированного по возрастанию, а в третьей строке — k вопросов. Все элементы целые, в диапазоне $[-10^9; 10^9]$.

Формат выходных данных

Для каждого из k чисел второго массива выведите в отдельную строку «YES», если это число встречается в первом массиве, и «NO» в противном случае.

Пример

стандартный ввод	стандартный вывод
10 5	NO
1 2 3 4 5 6 7 8 9 10	NO
-2 0 4 9 12	YES YES NO

Замечание

В массиве чисел от 1 до 10 есть числа 4, 9, но нет чисел -2, 0, 12

Задача В. Рядом

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вам дан отсортированный массив a_n и запросы для поиска элемента, максимально близкого к запрошенному x ($|a_i - x| \rightarrow \min$). Если есть несколько значений с минимальной разницей по модулю, надо вывести минимальное.

Формат входных данных

В первой строке входных данных содержатся числа n и k ($1 \leq n, k \leq 10^5$). Во второй строке задаются n чисел первого массива, отсортированного по неубыванию, а в третьей строке — k запросов. Каждое число находится в промежутке [−2 · 10⁹; 2 · 10⁹].

Формат выходных данных

Для каждого из k чисел выведите в отдельную строку число из первого массива, наиболее близкое к данному. Если таких несколько, выведите меньшее из них.

Пример

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

Замечание

В примере дан массив 1, 3, 5, 7, 9.

Поскольку $|2 - 1| = |2 - 3|$, то ответ на первый запрос — 1 (как $\min(1, 3)$) Аналогично на 4, 8 ответы 3, 7.

Число 1 присутствует в массиве, поэтому в данном случае мы найдем не просто ближайшее число, а 1, ведь $|1 - 1| = 0$

Задача С. Отгадай число

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Эта задача немного необычна — в ней вам предстоит реализовать интерактивное взаимодействие с тестирующей системой. Это означает, что вы можете делать запросы и получать ответы в online-режиме. Обратите внимание, что ввод/вывод в этой задаче — стандартный (то есть с экрана на экран). После вывода очередного запроса обязательно используйте функции очистки потока, чтобы часть вашего вывода не осталась в каком-нибудь буфере. Например, на C++ надо использовать функцию `fflush(stdout)`, на Java вызов `System.out.flush()`, на Pascal `flush(output)` и `stdout.flush()` для языка Python.

В этой задаче вам предстоит в интерактивном режиме угадать число x , которое загадала тестирующая система. Про загаданное число x известно, что оно целое и лежит в границах от 1 до n включительно (значение n известно заранее).

Вы можете делать запросы к тестирующей системе, каждый запрос — это вывод одного целого числа от 1 до n . Есть два варианта ответа тестирующей системы на запрос:

- строка «<» (без кавычек), если загаданное число меньше числа из запроса;
- строка «>=» (без кавычек), если загаданное число больше либо равно числу из запроса.

В случае, если ваша программа наверняка угадала нужное число x , выведите строку вида «! x », где x — это ответ, и завершите работу своей программы.

Вашей программе разрешается сделать не более 25 запросов.

Формат входных данных

Для чтения ответов на запросы программа должна использовать стандартный ввод.

В первой строке входных данных будет содержаться целое положительное число n ($1 \leq n \leq 10^6$) — максимально возможное число, которое может быть загадано.

В следующих строках на вход вашей программе будут подаваться строки вида «<» и «>=». i -я из этих строк является ответом системы на ваш i -й запрос. После того, как ваша программа угадала число, выведите «! x » (без кавычек), где x — это ответ, и завершите работу своей программы.

Тестирующая система даст вашей программе прочитать ответ на запрос из входных данных только после того, как ваша программа вывела соответствующий запрос системе и выполнила операцию `flush`.

Формат выходных данных

Для осуществления запросов программа должна использовать стандартный вывод.

Ваша программа должна выводить запросы — целые числа x_i ($1 \leq x_i \leq n$) по одному в строке (не забывайте выводить «перевод строки» после каждого значения x_i). После вывода каждой строки программа должна выполнить операцию `flush`.

Каждое из значений x_i обозначает очередной запрос к системе. Ответ на запрос программа сможет прочесть из стандартного ввода. В случае, если ваша программа угадала число x , выведите строку вида «! x » (без кавычек), где x — ответ, и завершите работу программы.

Примеры

стандартный ввод	стандартный вывод
20	
<	5
>=	3
>=	4
	! 4

Замечание

Вот заготовка для python

```
import sys
n = int(input())

def query(x):
    print(x)
    sys.stdout.flush()
    return input()
```

Задача D. Квадратный корень и квадратный квадрат

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Найдите такое число x ($x \geq 0$), что $x^2 + \sqrt{x+1} = C$, с точностью не менее 6 знаков после точки.

Формат входных данных

В единственной строке содержится вещественное число $1 \leq C \leq 10^{10}$.

Формат выходных данных

Выведите одно число — искомый x .

Примеры

стандартный ввод	стандартный вывод
2.0000000000	0.80926547401163950735
18.0000000000	3.97119409286392421876

Задача Е. Корень кубического уравнения

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Дано кубическое уравнение $ax^3 + bx^2 + cx + d = 0$ ($a \neq 0$). Известно, что у этого уравнения ровно один корень. От вас требуется его найти.

Заметьте, что разрешены различные случаи: любой из коэффициентов может быть положительным, отрицательным, или все коэффициенты, кроме a , могут быть равны нулю.

Формат входных данных

Во входных данных через пробел записаны четыре целых числа: $-1000 \leq a, b, c, d \leq 1000$.

Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 4 знаков после десятичной точки.

Пример

стандартный ввод	стандартный вывод
1 -3 3 -1	1.0000005398739177931

Задача F. Сортировка слиянием с приколом

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Это задача требует от вас написать сортировку слиянием для за заданного массива $A = \langle a_1, a_2, \dots, a_n \rangle$.

Разумеется, все не так просто. Мы хотим, чтобы вы не смогли сдать какой-нибудь `a.sort()`. А значит, мы попросим от вас посчитать количество инверсий в массиве. Для этого вам потребуется модифицировать стадию слияния массивов — попробуйте понять, что происходит с числом инверсий при слиянии двух массивов. Мы подскажем, что в отсортированном массиве 0 инверсий, а операция `merge` принимает два отсортированных массива и сливает их в один большой отсортированный массив.

Количество инверсий — это количество пар (i, j) таких, что $i < j$ и $a_i > a_j$. Обратите внимание на то, что ответ может не влезать в 32-битный тип данных, если вы пишете не на python.

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов массива A — целых неотрицательных чисел, не превосходящих 10^9 .

Формат выходных данных

В первой строке выведите одно число — количество инверсий в массиве.

Во второй строке выведите отсортированный массив, элементы выводите через пробел.

Примеры

стандартный ввод	стандартный вывод
5 6 11 18 28 31	0 6 11 18 28 31
4 1 3 2 4	1 1 2 3 4

Задача G. Anti-qsrt test

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Рассмотрим алгоритм быстрой сортировки, с выбором в качестве барьераного элемента среднего элемента на отрезке ($q = a[(l + r) / 2]$):

```
void qsort(vector<int> &a, int left, int right)
// Сортировка a[left...right] включительно
{
    if (right <= left)
        return;
    int q = a[(l + r) / 2];
    int i = left;
    int j = right;
    while (i <= j) {
        while (a[i] < q)
            ++i;
        while (q < a[j])
            --j;
        if (i <= j) {
            swap(a[i], a[j]);
            ++i;
            --j;
        }
    }
    qsort(a, left, j);
    qsort(a, i, right);
}
```

По данному числу n составьте тест, являющийся перестановкой чисел от 1 до n , на котором этот алгоритм выполняет наибольшее число сравнений (подсчитываются сравнения $a[i] < q$ и $q < a[j]$).

Формат входных данных

Программа получает на вход одно целое число n , $1 \leq n \leq 70\,000$.

Формат выходных данных

Программа должна вывести перестановку чисел от 1 до n , на которой данная реализация алгоритма быстрой сортировки Хоара будет выполнять наибольшее число сравнений.

Можно вывести любой из возможных ответов.

Пример

стандартный ввод	стандартный вывод
3	1 3 2

Задача Н. Зеркальный код

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Недавно Петя достал с полки запылившуюся детскую энциклопедию и прочитал там про метод шифрования, который использовал сам Леонардо да Винчи. Метод относительно прост — писать зеркально. Ходят слухи, что для этого он просто прикладывал зеркало к краю страницы и писал, смотря в отражение.

Но Петя знает, что есть строки, которым такое шифрование ничего не сделает. Страна, которая одинаково читаются справа налево и слева направо называются палиндромами.

У Пети есть обычная строка, состоящая из больших букв латинского алфавита. Ему стало интересно, можно ли составить из каких-то букв этой строки палиндром. Петя хочет получить палиндром наибольшей длины, а из всех таких — первый в алфавитном порядке.

Так как Петя еще надо дочитать энциклопедию, то он попросил сделать это вас.

Формат входных данных

В первой строке входных данных содержится число N ($1 \leq N \leq 100000$). Во второй строке задается строка из N больших латинских букв (буквы записаны без пробелов).

Формат выходных данных

В единственной строке выходных данных выдайте искомый палиндром.

Примеры

стандартный ввод	стандартный вывод
3 AAB	ABA
6 QAZQAZ	AQZZQA
6 ABCDEF	A

Задача I. Что? Да! Пузырек

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам дан код сортировки пузырьком:

```
for iteration in range(n):
    had_swaps = False
    for i in range(n - 1):
        if a[i] > a[i + 1]:
            a[i], a[i + 1] = a[i + 1], a[i]
            had_swaps = True
    if not had_swaps:
        break
```

А еще дан массив размера n , состоящий из нулей. Все 0 по очереди превращаются в 1, в порядке, заданном в teste. От вас требуется после каждой замены говорить, сколько внешних циклов сделает алгоритм сортировки (до вызова break), если его запустить на массиве.

Заметьте, что вы не должны сортировать, вы должны только сказать, сколько итераций понадобилось бы на сортировку.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 300000$) — количество элементов в массиве.

Следующая строка содержит n целых различных чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — позиции нулей, если смотреть слева направо, которые меняются на 1. Сначала меняется ноль, находящийся на позиции p_1 , затем на позиции p_2 и так далее.

Формат выходных данных

Выведите $n + 1$ число a_0, a_1, \dots, a_n , где a_0 — количество итераций для упорядочивания последовательности в начале, a_1 — сложность упорядочивания после первой замены и так далее.

Примеры

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

Замечание

В первом teste сначала нужно отсортировать $[0, 0, 0, 0]$, это можно сделать за один проход (который поймет, что все отсортировано).

Затем нужно отсортировать $[1, 0, 0, 0]$, это можно сделать за 2 прохода. Первый проход сделает все свапы, а второй проверит, что большие свапы не нужны.

Затем нужно отсортировать $[1, 0, 1, 0]$, на что потребуется 3 прохода.

Затем сортируется $[1, 0, 1, 1]$, на что потребуется 2 прохода - на первую единицу и финальная проверка сортированности.

Для сортировки $[1, 1, 1, 1]$ нужен один проход.

Задача J. Разбиение таблицы

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

Рассмотрим таблицу из n строк и m столбцов, в клетки которой по строкам записаны числа от 1 до $n \cdot m$. Сначала заполняется первая строка слева направо, затем вторая, и так далее. Другими словами, в клетку (r, c) записано число $(r - 1) \cdot m + c$.

На рисунке приведен пример такой таблицы для $n = 3$, $m = 5$.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Требуется разделить таблицу одним вертикальным или горизонтальным разрезом, проходящим по сторонам клеток, так чтобы сумма чисел в получившихся частях таблицы отличалась как можно меньше. В этой задаче в одном тесте вам придется ответить на несколько запросов об оптимальном разрезании таблицы.

Формат входных данных

В первой строке ввода задано целое число t — количество запросов ($1 \leq t \leq 10^5$).

В следующих t строках заданы по два числа n, m ($1 \leq n, m \leq 10^9$, $2 \leq n \cdot m \leq 10^9$).

Формат выходных данных

В t строках выведите ответы на запросы, по одному на строке.

Ответ на каждый запрос должен быть выведен в формате «D x », где D — это «V», если нужно резать по вертикали, «H» — если по горизонтали, а x — номер столбца или строки, перед которым надо сделать разрез. Строки пронумерованы от 1 до n , столбцы пронумерованы от 1 до m .

Если правильных ответов несколько, то надо вывести вариант с вертикальным разрезом, если он есть, а если и после этого вариантов несколько, то из вариантов с различными x следует выбрать тот, в котором x меньше.

Пример

стандартный ввод	стандартный вывод
5	V 3
1 3	V 5
4 7	V 8
1 10	H 3
3 3	V 4
3 5	