Задача 1. Бинарный поиск

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

В упорядоченном целочисленном массиве `методом бинарного поиска` найти заданное число и вывести его номер.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа N и X – размер массива и число, которое нужно найти в этом массиве ($1 \leqslant N \leqslant 10^6, -10^6 \leqslant X \leqslant 10^6$).

Во второй строке через пробел записаны N целых чисел в порядке неубывания – элементы массива. Все числа по модулю не превосходят 10^6.

Формат выходных данных

В выходной файл необходимо вывести номер элемента массива, содержащего заданное число X. Если такого числа нет, то вывести число -1.

Пример

input.txt	output.txt
5 9 2 4 7 9 12	3

Задача 2. Бинарный поиск 2

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

В упорядоченном массиве вещественных чисел `методом бинарного поиска` найти самое близкое число к заданному и вывести его номер.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа N и X – размер массива и число, близкое к которому нужно найти в этом массиве ($1 \leqslant N \leqslant 10^6, -10^6 \leqslant X \leqslant 10^6$).

Во второй строке через пробел записаны N вещественных чисел в порядке неубывания – элементы массива. Все числа по модулю не превосходят 10^7.

Формат выходных данных

В выходной файл необходимо вывести номер элемента массива, содержащего заданное число, значение которого наиболее близко к X. Если таких чисел больше одного, то вывести номер первого по порядку.

Пример

input.txt	output.txt
5 5 2.1 4.37 6.2 9.07 12.01	1

Задача 3. Поиск подстроки в строке

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

В файле дано две строки: первая строка T длины N и вторая строка P длины M.

Обе строки состоят только из маленьких букв латинского алфавита, причём $N, M \geqslant 1$ и $N + M \leqslant 10^6$.

Требуется найти в строке T все вхождения строки P как подстроки, и вывести позиции этих вхождений.

Позиция вхождения — это номер первого символа P в строке T (символы нумеруются начиная с нуля). Позиции нужно выводить в порядке возрастания.

Пример

input.txt	output.txt
abacababa aba	3 0 4 6

Пояснение к примеру

В примере имеется три вхождения шаблона: **aba**cababa (0), abac**aba**ba (4), abacab**aba** (6).

Задача 4. Бинарный поиск+

Источник:	повышенной сложности
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
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, соответсвующие 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 ($1 \leqslant N \leqslant 10^5$).

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

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

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 3 4 4 7 8 10 10 12 10 1 2 3 4 5 6 7 8 9 10	1 2 -1 2 5 -1 -1 5 -1 -1

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...