

Задача 4. Максимальный отрезок

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

Внимание: эта задача проверяется на **emailtester**.

Дан массив A длины N , все элементы массива A_i неотрицательные.

Далее нужно ответить на M вопросов. В каждом вопросе задана позиция L в массиве и число S . Нужно найти самый длинный отрезок с левым краем в L , сумма элементов которого не превышает S . Вывести требуется правый край R такого отрезка.

В данной задаче края отрезка задаются так, как принято в языке C, то есть:

1. Элементы массива нумеруются начиная с нуля.
2. Отрезок с краями $L < R$ включает L -ый элемент, но не включает R -ый элемент. Если $L = R$, то отрезок пустой.

В первой строке дано два целых числа: N — количество элементов массива A и M — количество вопросов ($1 \leq N, M \leq 10^5$).

В следующих N строках записаны элементы массива A_i , по одному в строке. Все эти числа целые неотрицательные и не превышают 10^9 .

В оставшихся M строках записаны вопросы, по одному в строке. Вопрос описывается двумя целыми числами: L — левый край искомого отрезка и S — насколько большой может быть сумма элементов отрезка ($0 \leq L \leq N, 0 \leq S \leq 10^{15}$).

Использование хедеров в этой задаче запрещается. Решение задачи должно быть разбито на три исходных файла: `sum.c`, `query.c` и `main.c`. Ровно эти файлы следует отправлять в систему тестирования.

В файле `sum.c` должны быть определены следующие функции:

```
//выполнить подготовку массива: запускается один раз перед запусками Sum
//здесь arr[0..n) -- это массив A из входного файла
void Init(const int *arr, int n);
//найти сумму элементов A на отрезке [l..r)
//использует данные, посчитанные функцией Init
int64_t Sum(int l, int r);
```

Используя эти функции, нужно реализовать в файле `query.c` ещё одну функцию:

```
//находит самый длинный отрезок с началом в l и суммой не более sum
//возвращает правый край искомого отрезка
int Query(int l, int64_t sum);
```

В файле `main.c` должна быть функция `main` (точка входа), чтение и запись данных, вызовы вышеописанных функций.

Можно использовать глобальные и статические переменные для хранения любых данных.

Пример

input.txt	output.txt
10 7	10
1	3
4	8
0	9
5	9
6	10
0	8
0	
1	
5	
3	
0 100	
0 5	
4 11	
4 12	
4 13	
10 100	
8 0	

Комментарий

Можно реализовать функцию `Init` за время $O(N)$, так что функция `Sum` будет работать за время $O(1)$. Для этого надо предподсчитать суммы на $[0, k)$ для всех k , а сумму на отрезке вычислять через две такие суммы.

Заметим, что сумма элементов массива на отрезке $[l, r)$ аналогична интегралу функции на отрезке $[a, b]$. В таком случае идея данной задачи является дискретным аналогом формулы Ньютона-Лейбница \Rightarrow)