

Задача А. Следующий

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

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

1. `add(i)` – добавить в множество S число i (если он там уже есть, то множество не меняется)
2. `next(i)` – вывести минимальный элемент множества, не меньший i . Если искомый элемент в структуре отсутствует, необходимо вывести -1 .

Формат входных данных

Исходно множество S пусто. Первая строка входного файла содержит n – количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо `«+ i»`, либо `«? i»`. Операция `«? i»` задает запрос `next(i)`.

Если операция `«+ i»` идет во входном файле в начале или после другой операции `«+»`, то она задает операцию `add(i)`. Если же она идет после запроса `«?»`, и результат этого запроса был y , то выполняется операция `add((i + y) mod 109)`.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Формат выходных данных

Для каждого запроса выведите одно число – ответ на запрос.

Пример

стандартный ввод	стандартный вывод
6	3
+ 1	4
+ 3	
+ 3	
? 2	
+ 1	
? 4	

Замечание

В этой задаче запрещено использовать STL.

Задача В. Персистентная очередь

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

Персистентные структуры данных поддерживают доступ и модификацию любой версии структуры данных. В этой задаче вам нужно реализовать персистентную очередь.

Очередь — это структура данных, которая хранит список целых чисел и поддерживает две операции: `push` и `pop`. Операция `push(x)` добавляет x в конец списка. Операция `pop` возвращает первый элемент списка и удаляет его.

В персистентной версии очереди каждая операция принимает дополнительный аргумент v . Изначально очередь имеет версию 0. Рассмотрим i -ю операцию над очередью. Если это `push(v, x)`, то число x добавляется в конец v -й версии очереди и в полученная очередь становится версии i (v -я версия остается неизменной). Если это `pop(v)`, то первое число удаляется из v -й версии очереди и полученная очередь становится версии i (так же, версия v остается неизменной).

Задана последовательность операций над персистентной очередью, выведите результаты всех операций `pop`.

Формат входных данных

Первая строка содержит целое число n — число операций ($1 \leq n \leq 200\,000$). Следующие n строк описывают операции. i -я из этих строк задает i -ю операцию. Операция `push(v, x)` задается, как “1 v x”, operation `pop(v)` задается, как “-1 v”. Гарантировано, что операция `pop` никогда не применяется к пустой очереди. Элементы, добавляемые в очередь, влезают в стандартный знаковый 32-битный целочисленный тип данных.

Формат выходных данных

Для каждой операции `pop` выведите значение элемента, который был извлечен из очереди.

Примеры

стандартный ввод	стандартный вывод
10	1
1 0 1	2
1 1 2	3
1 2 3	1
1 2 4	2
-1 3	4
-1 5	
-1 6	
-1 4	
-1 8	
-1 9	

Задача С. К-ый кратчайший путь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Вам дан произвольный взвешенный ориентированный граф на n вершинах и m рёбрах. Вершины пронумерованы целыми числами от 0 до $n - 1$.

Для каждого i от 1 до k , выведите длину i -го кратчайшего пути от вершины s до вершины t . Если такого пути не существует, выводите -1 . Пути считаются различными, если они образуют разные последовательности рёбер.

Формат входных данных

В первой строке задано пять целых чисел n, m, s, t, k ($1 \leq n, m, k \leq 3 \cdot 10^5$) ($0 \leq s, t < n$).

В следующих m строчках находятся по три числа u, v, c ($0 \leq u, v < n$) ($0 \leq c \leq 10^7$) — означающие, что граф содержит ребро из u в v веса c .

Формат выходных данных

Выведите k чисел, длины первых k кратчайших путей.

Пример

стандартный ввод	стандартный вывод
4 5 0 3 5	2
0 1 1	2
1 2 1	3
2 3 1	-1
0 2 1	-1
1 3 1	

Задача D. К-ая сумма

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Задан массив целых чисел $a_1, a_2, \dots, a_n$

Выведите k наименьших сумм среди всех непустых подпоследовательностей массива a .

Формат входных данных

Первая строка содержит два целых числа n, k ($1 \leq n \leq 200000, 1 \leq k \leq \min\{2^n - 1, 200000\}$).

Во второй строке находятся n целых чисел $a_1, a_2, \dots, a_n$ ($|a_i| \leq 10^9$).

Формат выходных данных

Выведите k целых чисел, задающие наименьшие суммы подпоследовательностей.

Примеры

стандартный ввод	стандартный вывод
2 3 -1 1	-1 0 1
3 7 -1 0 1	-1 -1 0 0 0 1 1