

## Задача А. Кубики

Имя входного файла: `cubes.in`  
Имя выходного файла: `cubes.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Одним прекрасным вечером, рассказывая очередную утиную историю своим любимым внукам — Билли, Дилли и Вилли, Скрудж МакДак вспомнил, как он любил играть с кубиками в детстве. Захваченный воспоминаниями, Скрудж предложил ребятам пособирать башенки из кубиков. Все с радостью поддержали его идею.

Всего утятка собрали  $n$  башенок. В  $i$ -й башенке оказалось  $a_i$  кубиков, поставленных друг на друга. Скрудж заметил, что башенки имеют разную высоту. Ему, как большому любителю порядка, это не понравилось, и он решил исправить ситуацию. Скрудж решил, что он будет перекладывать, добавлять и убирать кубики так, чтобы все башенки оказались одинаковой высоты. За одно действие Скрудж может переложить кубик с одной башенки на другую, убрать кубик из конструкции, или взять кубик из набора и положить его на какую-нибудь башенку. Кубиков в наборе неограниченное количество. Высота башенки определяется как количество кубиков в ней.

Помогите Скруджу посчитать, какое минимальное количество действий ему понадобится для того, чтобы сделать все башенки одинаковой высоты!

### Формат входных данных

В первой строке входного файла даны два числа  $n$  ( $1 \leq n \leq 1000$ ) — количество башенок. Во второй строке входного файла дано  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 1000$ ) — количество кубиков в  $i$ -й башенке.

### Формат выходных данных

В единственной строке выходного файла выведите единственное число — ответ на задачу.

### Пример

<code>cubes.in</code>	<code>cubes.out</code>
5	
3 2 2 5 4	3

### Система оценки

В этой задаче 25 тестов, каждый тест оценивается независимо в 4 балла.

## Задача В. Очередь в банк

Имя входного файла:	<code>longqueue.in</code>
Имя выходного файла:	<code>longqueue.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Недавно Скрудж устроился работать охранником в банке. Работа скучная, делать нечего, поэтому он начал следить за очередью. Исходно в очереди стоит  $n$  человек. Так как до этого несколько лет Скрудж работал психологом, он смог довольно точно оценить настроение каждого человека в очереди. Скрудж пронумеровал людей в очереди по порядку, начиная с нуля, таким образом получилось, что номер человека в очереди равен числу людей, которое стоит в очереди перед ним. Настроение  $i$ -го человека он описал целым неотрицательным числом  $a_i$ . Скрудж считает, что у человека хорошее настроение, если оно не меньше  $x$ . Если это не так, то настроение у человека плохое.

Люди приходят в очередь, уходят из нее. Если в очередь приходит новый человек, Скрудж мгновенно оценивает его настроение, и с течением времени оно не меняется.

Теперь Скрудж придумал себе следующее занятие: в некоторые моменты времени он выбирает одного человека из очереди и считает, сколько перед ним стоит человек с хорошим настроением. Это занятие уже показалось ему интересным, и он решил придумать, как его можно автоматизировать. Так как сам Скрудж не силен в программировании, помочь в решении этой задачи он попросил у вас. Помогите ему!

### Формат входных данных

В первой строке входного файла даны два числа  $n, x$  ( $1 \leq n \leq 100\,000, 0 \leq x \leq 10^9$ ) — начальное количество человек в очереди и нижняя граница хорошего настроения.

В следующей строке даны  $n$  чисел  $a_i$  — настроения людей в очереди ( $0 \leq a_i \leq 10^9$ ).

В третьей строке входного файла дано число  $m$  ( $1 \leq m \leq 100\,000$ ) — количество событий, которые происходили с очередью. В следующих  $m$  строках дано описание событий. Событие описывается одним из трех способов:

- 1  $a$  ( $0 \leq a \leq 10^9$ ) — в конец очереди приходит человек с настроением, равным  $a$ .
- 2 — из очереди уходит человек, перед которым никого не стоит (в нумерации Скруджа он имеет номер 0). После этого Скрудж мысленно уменьшает номера людей в очереди на 1.
- 3  $i$  — Скрудж хочет узнать, сколько людей с хорошим настроением стоит перед человеком, перед которым в очереди в этот момент стоит  $i$  человек.

Гарантируется, что все запросы корректны: если в очереди никого нет, то операция второго типа не выполняется, а количество человек в очереди всегда будет строго больше  $i$  в запросе третьего типа.

### Формат выходных данных

На каждый запрос третьего типа в отдельной строке выходного файла выведите одно число — количество человек с хорошим настроением, которые стоят перед человеком с данным номером.

## Пример

longqueue.in	longqueue.out
1 2	0
3	1
5	2
1 2	
1 1	
3 0	
3 1	
3 2	
2 2	0
1 2	0
7	0
3 0	0
3 1	1
2	
3 0	
1 3	
3 0	
3 1	

## Система оценки

Первая группа тестов состоит из тестов, для которых выполняется ограничение  $n, m \leq 1\,000$ . Стоимость группы составляет 40 баллов.

Вторая группа тестов состоит из тестов, для которых выполняется ограничение  $n, m \leq 100\,000$ . Стоимость группы составляет 60 баллов.

## Задача С. Выборы президента

Имя входного файла: parties.in  
Имя выходного файла: parties.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Мистер Скрудж — председатель местного парламента. На повестке дня в парламенте выбор президента. Конечно, президентом нужно выбрать одного из парламентариев. В парламенте есть две фракции — Даки и Маусы. Членам обеих партий запрещено голосовать за политиков, которые состоят в той же партии, что и голосующий. Председатель парламента, то есть, Скрудж, не имеет права голоса, и не может быть избран президентом.

В процессе выборов каждый из парламентариев проголосовал ровно за одного парламентария, состоящего в другой партии. После этого список, в котором указано, сколько голосов получил каждый из политиков, был передан Скруджу.

Получив список, Скрудж понял, что он не знает, в какой партии состоит какой кандидат, а это важно. Помогите ему составить какое-либо распределение политиков по партиям, удовлетворяющее всем условиям, или выясните, что список, который получил Скрудж, некорректен.

### Формат входных данных

В первой строке входного файла дано целое число  $n$  ( $2 \leq n \leq 100\,000$ ) — число парламентариев.

В следующей строке дано  $n$  целых чисел  $a_i$  ( $0 \leq a_i < n$ , сумма всех  $a_i$  равна  $n$ ) — число голосов, которые получил  $i$ -й парламентарий.

### Формат выходных данных

В первой строке выходного файла выведите «YES», если можно таким образом назначить каждому парламентарию его партию, и выбрать, за кого он проголосовал, чтобы список, полученный Скруджем, оказался корректен, и «NO» в противном случае.

Если искомое назначение существует, то во второй строке выведите  $n$  целых чисел — 1, если  $i$ -й политик состоит в партии Даков, или 2, если он состоит в партии Маусов.

### Пример

parties.in	parties.out
5	YES
1 2 0 2 0	1 1 2 2 2

### Система оценки

Первая группа тестов состоит из тестов, для которых выполняется ограничение  $1 \leq n \leq 15$ . Стоимость группы составляет 20 баллов.

Вторая группа тестов состоит из тестов, для которых выполняется ограничение  $1 \leq n \leq 300$ . Стоимость группы составляет 40 баллов.

Третья группа тестов состоит из тестов, для которых выполняется ограничение  $1 \leq n \leq 100\,000$ . Стоимость группы составляет 40 баллов.

## Задача D. Пирожные

Имя входного файла: `cakes.in`  
Имя выходного файла: `cakes.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Сегодня у Скруджа день рождения!

В подарок он получил целый стол пирожных. Так как у миллионеров не очень много свободного времени, Скрудж хочет съесть как можно больше пирожных за  $T$  секунд.

Стол с пирожными можно представить как бесконечную прямую. Каждое пирожное задается на этой прямой своей координатой  $x_i$ . Для того, чтобы перейти от пирожного  $i$  к пирожному  $j$  Скрудж тратит  $|x_i - x_j|$  секунд. Также, для каждого пирожного Скрудж прикинул время  $t_i$  в секундах, за которое он сможет его съесть. Если несколько пирожных располагаются в одной точке, то Скруджу не надо перемещаться от одного к другому, но он может есть их только по очереди.

Изначально Скрудж стоит в точке с координатой 0. Помогите Скруджу выяснить какое максимальное количество пирожных он может успеть съесть за время  $T$ .

### Формат входных данных

В первой строке входного файла давно два целых числа  $n$  и  $T$  ( $1 \leq n \leq 100\,000$ ,  $1 \leq T \leq 10^9$ ) — количество пирожных и доступное время.

В каждой из следующих  $n$  строк дано по два целых числа  $x_i$  и  $t_i$  ( $1 \leq x_i, t_i \leq 10^9$ ) — координата  $i$ -го пирожного и время, за которое Скрудж может его съесть. Пирожные даны в порядке неубывания координаты, то есть для любых  $i$  и  $j$ , таких, что  $i < j$  верно, что  $x_i \leq x_j$ .

### Формат выходных данных

В единственной строке выходного файла выведите максимальное количество пирожных, которые Скрудж может успеть съесть за время  $T$ .

### Пример

<code>cakes.in</code>	<code>cakes.out</code>
3 10 1 4 2 5 3 3	2
3 10 1 2 2 2 3 3	3
8 100 1 21 3 10 4 3 5 19 8 8 9 32 50 1 100 1	5

### Замечание

В первом примере Скруджу нужно перейти от точки с координатой 0 к точке с координатой 1, съесть первое пирожное, потом перейти к точке с координатой 3 и съесть третье пирожное.

### Система оценки

Первая группа тестов состоит из тестов, для которых выполняются ограничения  $1 \leq n \leq 20$ ,  $1 \leq T \leq 10^9$ . Стоимость группы составляет 15 баллов.

Вторая группа тестов состоит из тестов, для которых выполняются ограничения  $1 \leq n \leq 1\,000, 1 \leq T \leq 1\,000$ . Стоимость группы составляет 20 баллов.

Третья группа тестов состоит из тестов, для которых выполняются ограничения  $1 \leq n \leq 1\,000, 1 \leq T \leq 10^9$ . Стоимость группы составляет 25 баллов.

Четвертая группа тестов состоит из тестов, для которых выполняются ограничения  $1 \leq n \leq 100\,000, 1 \leq T \leq 10^9$ . Стоимость группы составляет 40 баллов.

## Задача Е. Счета дядюшки Скруджа

Имя входного файла: accounts.in  
Имя выходного файла: accounts.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Дядюшка Скрудж известен своей жадностью. Когда-то давно он завел аж  $n$  банковских счетов. На каждый из этих счетов ежедневно приходит фиксированная положительная сумма в  $b_i$  долларов. Таким образом, через  $t$  дней на  $i$ -м счету находится  $t \times b_i$  долларов.

Сегодня Билли, Вилли и Дилли решили проучить своего дедушку. Они украли из его бухгалтерии всю информацию о числах  $b_i$ . Теперь дядюшка Скрудж даже не сможет посмотреть сумму на каждом из счетов, ведь число  $b_i$  является также паролем для  $i$ -го из них. Но ребята понимают, что это слишком жестокая шутка над Скруджем, поэтому они решили дать ему шанс и сделали  $t$  подсказок.

Каждая подсказка состоит в следующем: ребята выбирают какой-либо день, некоторое множество счетов и сообщают Скруджу номера этих счетов и сумму на них в этот день. Номер дня ребята ему не говорят.

Теперь Скрудж может попытаться восстановить числа  $b_i$  по имеющимся подсказкам. Он в отчаянии: для него это слишком сложная задача, поэтому он попросил вас помочь ему.

### Формат входных данных

В первой строке входного файла даны два целых числа  $n, m$  ( $1 \leq n \leq 100\,000, 1 \leq m \leq 100\,000$ ) — количество счетов дядюшки Скруджа и количество подсказок Билли, Вилли и Дилли. В следующих  $2m$  строках описаны подсказки дяде Скруджу: сначала идет целое число  $k_i$  ( $1 \leq k_i \leq n$ ) — количество счетов, для которых ребята записали их состояние в некоторый день, в следующей строке идут  $2k_i$  целых чисел  $c_{i,j}, x_{i,j}$  ( $1 \leq c_{i,j} \leq n, 1 \leq x_{i,j} \leq 10^{18}$ ) — номер счета и сумма на нем. Гарантируется, что для каждой подсказки  $c_{i,j}$  различны.

Гарантируется, что сумма  $k_i$  не превосходит  $3 \cdot 10^5$ .

### Формат выходных данных

В первой строке выведите «NO», если решения не существует и «YES» в противном случае. Если решение существует, следующая строка должна содержать  $n$  целых чисел  $b_i$  ( $1 \leq b_i \leq 10^{18}$ ) — сколько долларов приходит на  $i$ -й счет ежедневно. Если решений, удовлетворяющих всем подсказкам, несколько, выведите любое.

### Пример

accounts.in	accounts.out
3 2 2 1 6 2 9 1 1 8	YES 2 3 1
5 3 2 1 4 2 6 2 2 12 3 8 3 2 9 3 6 4 3	YES 2 3 2 1 1
3 2 2 1 6 2 9 2 1 9 2 6	NO

## Замечание

В первом примере номер дня для первой подсказки равен 3, для второй 4, считая от дня открытия счетов. Во втором примере номер дня для первой подсказки равен 2, для второй 4, для третьей 3, считая от дня открытия счетов. В третьем примере решения не существует.

## Система оценки

Первая группа тестов состоит из тестов, для которых выполняется ограничение  $x_{i,j} \leq 100\,000$ . Стоимость группы составляет 50 баллов.

Вторая группа тестов состоит из тестов, для которых выполняются полные ограничения. Стоимость группы составляет 50 баллов.