

## Задача А. Игра с шариками

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

В новой игре с шариками у игрока есть мешок с  $n$  шариками, на  $i$ -м из которых написано число  $a_i$ . Игрок выбирает несколько шариков из мешка. Если он вынул шарик с суммой чисел, равной  $s$ , то он получает  $|s|$  очков ( $|x|$  обозначает абсолютное значение  $x$ , то есть  $x$  при  $x > 0$  и  $-x$  при  $x < 0$ ).

Определите наибольшее количество очков, которое может получить игрок.

### Формат входных данных

В первой строке ввода находится единственное целое число  $n$  ( $1 \leq n \leq 10^5$ ).

В следующей строке находятся  $n$  целых чисел, написанных на шариках ( $-10^9 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите единственное число — наибольшее количество очков, которое может получить игрок.

### Система оценки

Решения, корректно работающие при  $n, |a_i| \leq 100$ , будут набирать не менее 25 баллов.

Решения, корректно работающие при  $|a_i| \leq 100$ , будут набирать не менее 50 баллов.

### Пример

стандартный ввод	стандартный вывод
3 2 -3 2	4

### Замечание

В примере наиболее выгодно вытащить из мешка шарик со значениями 2, 2. В мешке останется шар со значением  $-3$ , а выигрыш игрока составит 4.

## Задача В. Flappy Bird

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

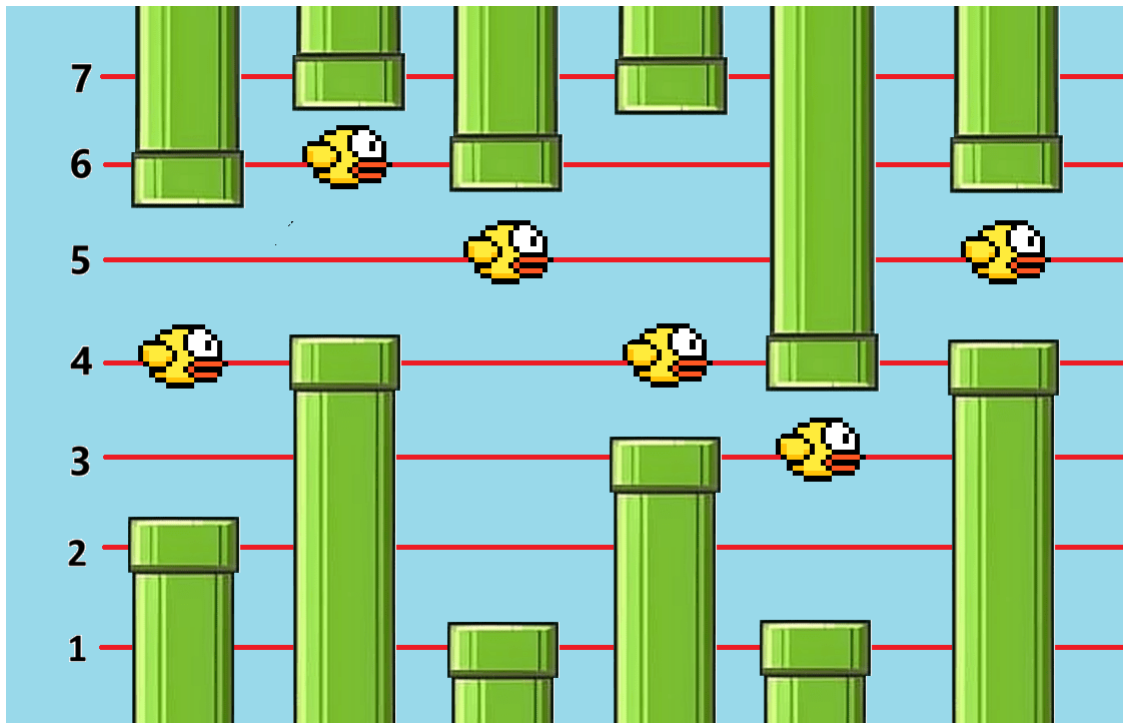
Антон играет в игру Flappy Bird. Антон играет за птичку, которой нужно пролететь по уровню слева-направо. Высота уровня равна  $h$ , то есть есть  $h$  различных высот, на которых может находиться птичка. Высоты пронумерованы числами от 1 до  $h$  от самой низкой до самой высокой. Уровень состоит из  $n$  столбцов, расположенных друг за другом слева направо, и пронумерованных числами от 1 до  $n$ . В  $i$ -м столбце есть две трубы, одна расположена снизу, другая сверху. Нижняя труба занимает все высоты от 1 до  $l_i$  включительно, а верхняя занимает все высоты от  $r_i$  до  $h$  включительно. Это значит, что в  $i$ -м столбце птичка может находиться на высотах от  $l_i + 1$  до  $r_i - 1$  включительно.

Птичка летит следующим образом: изначально она находится в первом столбце на высоте  $s$  ( $l_1 < s < r_1$ ), а затем  $n - 1$  раз перемещается в следующий столбец. Перед перемещением в следующий столбец игрок может нажать кнопку прыжка, тогда птичка поднимется на  $k$  вверх, т.е. если птичка в текущем столбце находилась на высоте  $x$ , тогда в следующем столбце она окажется на высоте  $x + k$ . Если же игрок не нажмет кнопку прыжка, то птичка снизится на одну единицу высоты, т.е. если она была на высоте  $x$ , то окажется на высоте  $x - 1$ . Заметим, что вертикальное перемещение птички происходит между столбцами. Это значит, что внутри одного столбца птичка проходит только по одной высоте и никак не перемещается горизонтально. Единственное ограничение для ее положения — ее высота в  $i$ -м столбце должна быть от  $l_i + 1$  до  $r_i - 1$  включительно.

Отметим также, что птичка никогда не может опускаться ниже первой и подниматься выше  $h$ -й высоты.

Уровень будет пройден, если птичка дойдет до  $n$ -го столбца.

Рассмотрим пример уровня (это первый тестовый случай в тесте из условия):



Здесь горизонтальные линии обозначают высоты, на которых может находиться птичка, они пронумерованы от 1 до 7 (т. е.  $h = 7$ ). Изначально птичка находится на высоте  $s = 4$ , высота ее прыжка равна  $k = 2$ . В данном примере существует ровно один способ пройти уровень, и он изображен на рисунке. В каждом столбце птичка нарисована на той высоте, на которой она должна находиться в этом столбце.

Помогите Антону и узнайте, может ли он пройти уровень (т. е. может ли птичка двигаться так, чтобы добраться до  $n$ -го столбца).

### Формат входных данных

В первой строке дано одно целое число  $t$  ( $1 \leq t \leq 10\,000$ ) — количество наборов входных данных. Далее следуют описания каждого из наборов входных данных.

В первой строке каждого набора входных данных дано четыре целых числа  $n$ ,  $h$ ,  $k$  и  $s$  ( $2 \leq n \leq 100\,000$ ,  $4 \leq h \leq 10^9$ ,  $1 \leq k \leq h - 3$ ,  $2 \leq s \leq h - 1$ ) — количество столбцов, высота уровня, высота прыжка и стартовая высота.

В следующих  $n$  строках описываются уровни. В  $i$ -й из них дано два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i < r_i \leq h$ ,  $l_i + 2 \leq r_i$ ) — высоты нижней и верхней трубы в  $i$ -м столбце. Гарантируется, что  $l_1 < s < r_1$ .

Пусть  $N$  равно сумме  $n$  по всем наборам входных данных. Гарантируется, что  $N \leq 500\,000$ .

### Формат выходных данных

Выведите  $t$  строк. В  $i$ -й строке выведите **Yes** если Антон может пройти уровень из  $i$ -го набора входных данных, иначе выведите **No**.

### Система оценки

Всего в этой задаче 20 тестов (не считая теста из условия). Оценка потестовая, каждый из тестов, кроме теста из условия, оценивается в 5 баллов. На некоторые из тестов наложены дополнительные ограничения, указанные в таблице.

Тесты	Дополнительные ограничения	Комментарий
1	—	Тест из условия
2, 3, 4	$r_i = l_i + 2$	—
5, 6, 7	$n \leq 10$	—
8, 9, 10	$h \leq 100$	—
11, 12, 13	$r_i - l_i \leq 100$	—
14, 15, 16	$n \leq 1000, N \leq 30\,000$	—
17, 18	$k = 1$	—
19, 20, 21	—	—

## Пример

стандартный ввод	стандартный вывод
4	Yes
6 7 2 4	No
2 6	Yes
4 7	No
1 6	
3 7	
1 4	
4 6	
5 7 1 4	
1 7	
2 6	
1 7	
3 5	
1 7	
8 100 10 50	
49 51	
48 50	
58 60	
68 70	
78 80	
77 79	
87 89	
86 88	
7 8 5 7	
1 8	
1 8	
1 8	
1 8	
1 8	
1 8	
1 8	
1 7	

## Замечание

Первый тестовый случай в примере совпадает с уровнем на картинке.

## Задача С. Иван и дома

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

Иван коренной житель города Дубкинцово, и недавно он стал мэром своего города. В Дубкинцово есть большая улица, на которой в один ряд расположены  $n$  домов. Высота  $i$ -го по счету дома на этой улице равна  $a_i$  метров.

Население Дубкинцово постоянно растет, и управлять городом становится сложнее. В целях оптимизации управления Иван принял решение разбить улицу на участки, состоящие из подряд идущих домов. При этом каждый дом должен относиться ровно к одному участку.

По результатам голосования жителей на онлайн-портале города было принято решение, что участки улицы должны обладать следующим свойством: для любого подотрезка подряд идущих домов на одном участке произведение их высот не должно являться полным квадратом некоторого целого числа. Т.е. если участок состоит из домов с номерами от  $l$  до  $r$ , то не существует таких  $i$  и  $j$  ( $l \leq i \leq j \leq r$ ), что  $\prod_{k=i}^j a[k] = x^2$  для любого  $x$ . Известно, что в Дубкинцово не строят домов с высотой, равной полному квадрату некоторого числа, и поэтому улицу гарантированно можно разбить на такие участки.

Ивану необходимо разработать план разбиения улицы на участки так, чтобы это условие было выполнено. Для экономии бюджета Иван хочет разделить улицу на минимальное количество участков.

### Формат входных данных

В первой строке дано одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество домов.

В следующей строке дано  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $2 \leq a_i \leq 10^5$ ) — высоты домов в порядке следования. Гарантируется, что никакое  $a_i$  не является полным квадратом целого числа.

### Формат выходных данных

Выведите одно целое число — минимальное количество участков, на которые можно разделить улицу, чтобы на каждом участке для любого подотрезка подряд идущих домов произведение их высот не являлось полным квадратом некоторого целого числа.

### Система оценки

В данной задаче 20 тестов, не считая тестов из условия. Каждый тест оценивается в 5 баллов.

Тесты	Дополнительные ограничения	Комментарии
1, 2	—	Тест из условия
3, 4, 5, 6	$n \leq 10, a_i \leq 10$	—
7, 8, 9, 10	$n \leq 10$	—
11, 12	$2 \leq a_i \leq 3$	—
13, 14, 15	—	Все $a_i$ простые числа
16, 17, 18	$a_i \leq 200$	—
19, 20, 21, 22	—	—

### Примеры

стандартный ввод	стандартный вывод
5 2 5 10 2 3	2
6 2 2 3 6 10 10	4

### Замечание

В первом примере можно разделить улицу на 2 участка:  $[2, 5]$  и  $[10, 2, 3]$ .

На первом участке будет три подотрезка из подряд идущих домов:  $[2]$  — произведение высот домов равно 2, это не полный квадрат;  $[5]$  — произведение высот домов равно 5, это не полный квадрат;  $[2, 5]$  — произведение равно 10, это не полный квадрат.

На втором участке будет шесть подотрезков из подряд идущих домов:  $[10]$ ,  $[2]$ ,  $[3]$ ,  $[10, 2]$ ,  $[2, 3]$ ,  $[10, 2, 3]$ . Произведение высот домов на каждом из отрезков не является полным квадратом.

Можно показать, что на меньшее количество участков разделить улицу нельзя.

## Задача D. Прятки на дереве

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

Бельчата решили поиграть в прятки на дереве. Правила игры просты: выбирается вершина  $v$  — в ней будет находиться водящий бельчонок. Остальные бельчата разбегаются по дереву и прячутся в его вершинах. Чем дальше вершина от водящего, тем она безопаснее.

Изначально дерево состоит из единственной вершины с номером 1. Вам нужно обрабатывать запросы двух видов:

- $+ u$  — создание новой вершины (ее номер будет на единицу больше текущего размера дерева), которая присоединяется к вершине  $u$
- $? u$  — пусть водящий бельчонок находится в вершине  $u$ . Нужно вычислить расстояние до самой безопасной вершины дерева, а также количество таких вершин

Вершина  $u_i$  в запросе  $i$  вычисляется по формуле

$$u_i = (ans_{i-1} \cdot m + v_i - 1) \bmod sz + 1$$

где  $ans_{i-1}$  — сумма наибольшего расстояния и количества вершин из ответа на предыдущий запрос типа  $?$  (положим  $ans_0 = 0$ ),  $v_i$  — параметр запроса из ввода,  $sz$  — текущий размер дерева.

### Формат входных данных

В первой строке ввода находятся два целых числа  $q$  и  $m$  — количество запросов и флаг, отвечающий за генерацию запросов ( $1 \leq q \leq 300\,000$ ,  $m \in \{0, 1\}$ ).

В следующих  $q$  строках находятся сами запросы в соответствии с форматом, описанным в условии, по одному в строке. Гарантируется, что  $0 \leq v_i \leq q$ .

### Формат выходных данных

В ответ на каждый запрос второго типа выведите на отдельной строке два числа, разделенные пробелом — расстояние от вершины  $u_i$  до самой дальней от нее вершины дерева, а также количество вершин на таком расстоянии от  $u_i$ .

### Система оценки

В задаче 28 тестов (помимо тестов из условия), оценка потестовая. Тесты задачи можно разбить на следующие подгруппы:

Макс. Балл	$m$	Дополнительные Ограничения	Комментарий
0		—	тест из условия
9	$m = 0$	$q \leq 100$	
20	$m = 0$	все запросы типа + идут до всех запросов типа ?	
8	$m = 0$	диаметр итогового дерева не превосходит 50	
16	$m = 1$	в запросах типа ? $u_i = 1$ (после пересчета по формуле)	
20	$m = 1$	в запросах типа ? $u_i = 1$ или $u_i$ — сосед вершины 1 (после пересчета по формуле)	
15	$m = 0$	—	
12	$m = 1$	—	

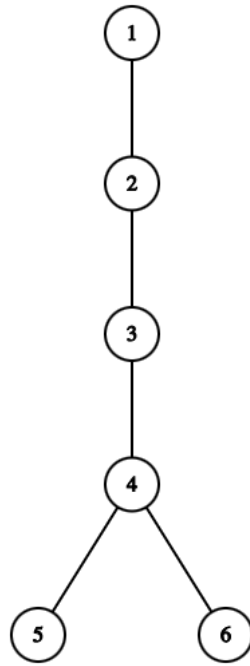
### Примеры

стандартный ввод	стандартный вывод
10 0 + 1 ? 2 + 2 ? 2 + 3 ? 2 + 4 ? 1 + 4 ? 1	1 1 1 2 2 1 4 1 4 2
10 1 + 1 ? 1 + 1 + 0 + 0 + 1 + 1 ? 1 ? 5 ? 2	1 1 3 2 3 2 4 2

### Замечание

Во втором тесте из условия значения  $u_i$  (после пересчета по формуле) следующие:  
[1, 1, 1, 2, 2, 3, 3, 3, 3, 7].

Также приведем итоговое дерево из первого примера:



## Задача А. Игра с шариками

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

В новой игре с шариками у игрока есть мешок с  $n$  шариками, на  $i$ -м из которых написано число  $a_i$ . Игрок выбирает несколько шариков из мешка. Если он вынул шарик с суммой чисел, равной  $s$ , то он получает  $|s|$  очков ( $|x|$  обозначает абсолютное значение  $x$ , то есть  $x$  при  $x > 0$  и  $-x$  при  $x < 0$ ).

Определите наибольшее количество очков, которое может получить игрок.

### Формат входных данных

В первой строке ввода находится единственное целое число  $n$  ( $1 \leq n \leq 10^5$ ).

В следующей строке находятся  $n$  целых чисел, написанных на шариках ( $-10^9 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите единственное число — наибольшее количество очков, которое может получить игрок.

### Система оценки

Решения, корректно работающие при  $n, |a_i| \leq 100$ , будут набирать не менее 25 баллов.

Решения, корректно работающие при  $|a_i| \leq 100$ , будут набирать не менее 50 баллов.

### Пример

стандартный ввод	стандартный вывод
3 2 -3 2	4

### Замечание

В примере наиболее выгодно вытащить из мешка шарик со значениями 2, 2. В мешке останется шар со значением  $-3$ , а выигрыш игрока составит 4.

## Задача В. Flappy Bird

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

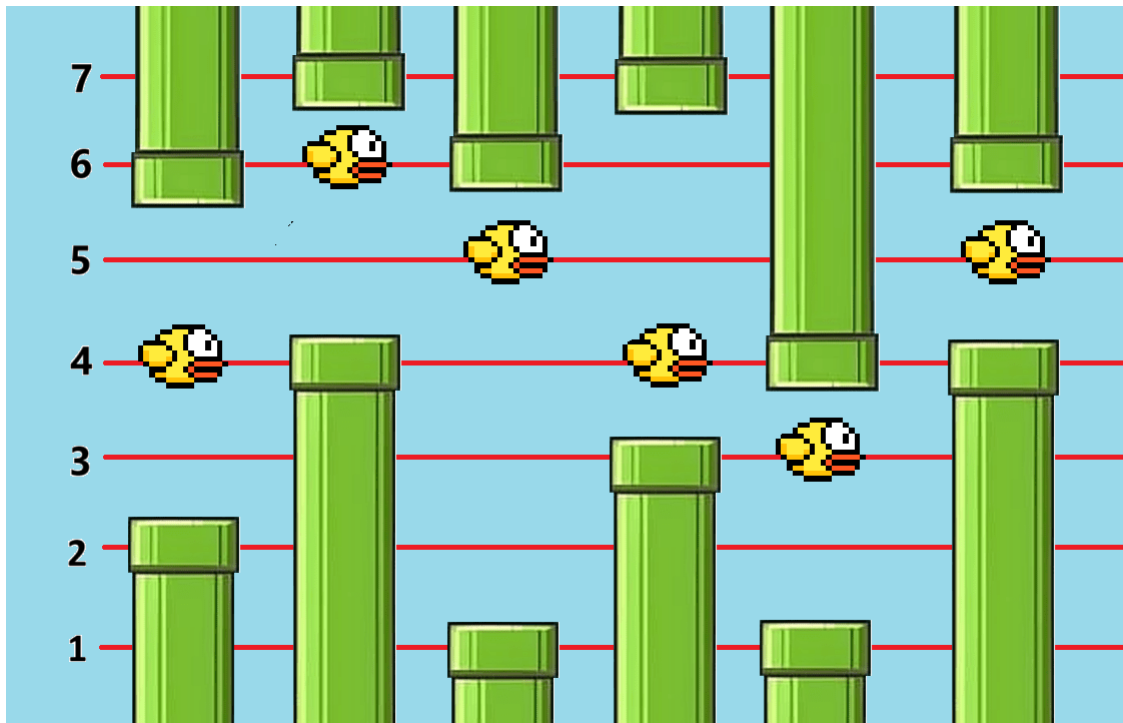
Антон играет в игру Flappy Bird. Антон играет за птичку, которой нужно пролететь по уровню слева-направо. Высота уровня равна  $h$ , то есть есть  $h$  различных высот, на которых может находиться птичка. Высоты пронумерованы числами от 1 до  $h$  от самой низкой до самой высокой. Уровень состоит из  $n$  столбцов, расположенных друг за другом слева направо, и пронумерованных числами от 1 до  $n$ . В  $i$ -м столбце есть две трубы, одна расположена снизу, другая сверху. Нижняя труба занимает все высоты от 1 до  $l_i$  включительно, а верхняя занимает все высоты от  $r_i$  до  $h$  включительно. Это значит, что в  $i$ -м столбце птичка может находиться на высотах от  $l_i + 1$  до  $r_i - 1$  включительно.

Птичка летит следующим образом: изначально она находится в первом столбце на высоте  $s$  ( $l_1 < s < r_1$ ), а затем  $n - 1$  раз перемещается в следующий столбец. Перед перемещением в следующий столбец игрок может нажать кнопку прыжка, тогда птичка поднимется на  $k$  вверх, т.е. если птичка в текущем столбце находилась на высоте  $x$ , тогда в следующем столбце она окажется на высоте  $x + k$ . Если же игрок не нажмет кнопку прыжка, то птичка снизится на одну единицу высоты, т.е. если она была на высоте  $x$ , то окажется на высоте  $x - 1$ . Заметим, что вертикальное перемещение птички происходит между столбцами. Это значит, что внутри одного столбца птичка проходит только по одной высоте и никак не перемещается горизонтально. Единственное ограничение для ее положения — ее высота в  $i$ -м столбце должна быть от  $l_i + 1$  до  $r_i - 1$  включительно.

Отметим также, что птичка никогда не может опускаться ниже первой и подниматься выше  $h$ -й высоты.

Уровень будет пройден, если птичка дойдет до  $n$ -го столбца.

Рассмотрим пример уровня (это первый тестовый случай в тесте из условия):



Здесь горизонтальные линии обозначают высоты, на которых может находиться птичка, они пронумерованы от 1 до 7 (т. е.  $h = 7$ ). Изначально птичка находится на высоте  $s = 4$ , высота ее прыжка равна  $k = 2$ . В данном примере существует ровно один способ пройти уровень, и он изображен на рисунке. В каждом столбце птичка нарисована на той высоте, на которой она должна находиться в этом столбце.

Помогите Антону и узнайте, может ли он пройти уровень (т. е. может ли птичка двигаться так, чтобы добраться до  $n$ -го столбца).

### Формат входных данных

В первой строке дано одно целое число  $t$  ( $1 \leq t \leq 10\,000$ ) — количество наборов входных данных. Далее следуют описания каждого из наборов входных данных.

В первой строке каждого набора входных данных дано четыре целых числа  $n$ ,  $h$ ,  $k$  и  $s$  ( $2 \leq n \leq 100\,000$ ,  $4 \leq h \leq 10^9$ ,  $1 \leq k \leq h - 3$ ,  $2 \leq s \leq h - 1$ ) — количество столбцов, высота уровня, высота прыжка и стартовая высота.

В следующих  $n$  строках описываются уровни. В  $i$ -й из них дано два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i < r_i \leq h$ ,  $l_i + 2 \leq r_i$ ) — высоты нижней и верхней трубы в  $i$ -м столбце. Гарантируется, что  $l_1 < s < r_1$ .

Пусть  $N$  равно сумме  $n$  по всем наборам входных данных. Гарантируется, что  $N \leq 500\,000$ .

### Формат выходных данных

Выведите  $t$  строк. В  $i$ -й строке выведите **Yes** если Антон может пройти уровень из  $i$ -го набора входных данных, иначе выведите **No**.

### Система оценки

Всего в этой задаче 20 тестов (не считая теста из условия). Оценка потестовая, каждый из тестов, кроме теста из условия, оценивается в 5 баллов. На некоторые из тестов наложены дополнительные ограничения, указанные в таблице.

Тесты	Дополнительные ограничения	Комментарий
1	—	Тест из условия
2, 3, 4	$r_i = l_i + 2$	—
5, 6, 7	$n \leq 10$	—
8, 9, 10	$h \leq 100$	—
11, 12, 13	$r_i - l_i \leq 100$	—
14, 15, 16	$n \leq 1000, N \leq 30\,000$	—
17, 18	$k = 1$	—
19, 20, 21	—	—

## Пример

стандартный ввод	стандартный вывод
4	Yes
6 7 2 4	No
2 6	Yes
4 7	No
1 6	
3 7	
1 4	
4 6	
5 7 1 4	
1 7	
2 6	
1 7	
3 5	
1 7	
8 100 10 50	
49 51	
48 50	
58 60	
68 70	
78 80	
77 79	
87 89	
86 88	
7 8 5 7	
1 8	
1 8	
1 8	
1 8	
1 8	
1 8	
1 8	
1 7	

## Замечание

Первый тестовый случай в примере совпадает с уровнем на картинке.

## Задача С. Иван и дома

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

Иван коренной житель города Дубкинцово, и недавно он стал мэром своего города. В Дубкинцово есть большая улица, на которой в один ряд расположены  $n$  домов. Высота  $i$ -го по счету дома на этой улице равна  $a_i$  метров.

Население Дубкинцово постоянно растет, и управлять городом становится сложнее. В целях оптимизации управления Иван принял решение разбить улицу на участки, состоящие из подряд идущих домов. При этом каждый дом должен относиться ровно к одному участку.

По результатам голосования жителей на онлайн-портале города было принято решение, что участки улицы должны обладать следующим свойством: для любого подотрезка подряд идущих домов на одном участке произведение их высот не должно являться полным квадратом некоторого целого числа. Т.е. если участок состоит из домов с номерами от  $l$  до  $r$ , то не существует таких  $i$  и  $j$  ( $l \leq i \leq j \leq r$ ), что  $\prod_{k=i}^j a[k] = x^2$  для любого  $x$ . Известно, что в Дубкинцово не строят домов с высотой, равной полному квадрату некоторого числа, и поэтому улицу гарантированно можно разбить на такие участки.

Ивану необходимо разработать план разбиения улицы на участки так, чтобы это условие было выполнено. Для экономии бюджета Иван хочет разделить улицу на минимальное количество участков.

### Формат входных данных

В первой строке дано одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество домов.

В следующей строке дано  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $2 \leq a_i \leq 10^5$ ) — высоты домов в порядке следования. Гарантируется, что никакое  $a_i$  не является полным квадратом целого числа.

### Формат выходных данных

Выведите одно целое число — минимальное количество участков, на которые можно разделить улицу, чтобы на каждом участке для любого подотрезка подряд идущих домов произведение их высот не являлось полным квадратом некоторого целого числа.

### Система оценки

В данной задаче 20 тестов, не считая тестов из условия. Каждый тест оценивается в 5 баллов.

Тесты	Дополнительные ограничения	Комментарии
1, 2	—	Тест из условия
3, 4, 5, 6	$n \leq 10, a_i \leq 10$	—
7, 8, 9, 10	$n \leq 10$	—
11, 12	$2 \leq a_i \leq 3$	—
13, 14, 15	—	Все $a_i$ простые числа
16, 17, 18	$a_i \leq 200$	—
19, 20, 21, 22	—	—

### Примеры

стандартный ввод	стандартный вывод
5 2 5 10 2 3	2
6 2 2 3 6 10 10	4

### Замечание

В первом примере можно разделить улицу на 2 участка:  $[2, 5]$  и  $[10, 2, 3]$ .

На первом участке будет три подотрезка из подряд идущих домов:  $[2]$  — произведение высот домов равно 2, это не полный квадрат;  $[5]$  — произведение высот домов равно 5, это не полный квадрат;  $[2, 5]$  — произведение равно 10, это не полный квадрат.

На втором участке будет шесть подотрезков из подряд идущих домов:  $[10]$ ,  $[2]$ ,  $[3]$ ,  $[10, 2]$ ,  $[2, 3]$ ,  $[10, 2, 3]$ . Произведение высот домов на каждом из отрезков не является полным квадратом.

Можно показать, что на меньшее количество участков разделить улицу нельзя.

## Задача D. Обмены рекордов

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Вам дан массив  $a_1, a_2, \dots, a_n$ , состоящий из целых чисел.

Назовем  $i$ -й элемент массива *рекордом*, если  $i > 1$  и выполняется  $a_i > \max\{a_1, a_2, \dots, a_{i-1}\}$ .

С массивом можно произвольное число раз производить следующую операцию:

- Выбрать любое такое  $i > 1$ , что  $i$ -й элемент массива  $a$  является рекордом.
- Поменять местами  $a_i$  и  $a_{i-1}$ .

Назовем массив  $b_1, b_2, \dots, b_n$  *хорошим*, если его можно получить из массива  $a_1, a_2, \dots, a_n$  применением описанной операции ноль или более раз.

Назовем *стоимостью* массива  $b_1, b_2, \dots, b_n$  сумму его префиксных сумм. Иными словами, стоимость массива  $b_1, b_2, \dots, b_n$  равна

$$\sum_{i=1}^n \sum_{j=1}^i b_j$$

Посчитайте сумму стоимостей всех хороших массивов по модулю  $10^9 + 7$ .

### Формат входных данных

В первой строке задано целое число  $n$  ( $1 \leq n \leq 5000$ ) – длина массива  $a$ .

Во второй строке заданы  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) – элементы массива  $a$ .

### Формат выходных данных

Выведите одно целое число – сумму стоимостей всех хороших массивов по модулю  $10^9 + 7$ .

### Система оценки

Обозначим через  $k$  количество рекордов в массиве  $a$ .

Задача состоит из 25 тестов, не считая тестов из условия. Каждый тест оценивается независимо в 4 балла.

Все тесты можно разделить на следующие группы:

Номер	Макс. Балл	Доп. ограничения
1	12	$n \leq 10$
2	12	$a_i > a_{i-1}$ для всех $i > 1$
3	8	$k \leq 1$
4	12	$k \leq 2$
5	16	Последние $k$ элементов массива являются рекордами
6	20	$n \leq 500$
7	20	–

### Примеры

стандартный ввод	стандартный вывод
3 2 1 3	38
5 1 3 1 4 4	294

## Замечание

В первом примере массив  $a$  равен  $[2, 1, 3]$ . Так как третий элемент является рекордом, то его можно поменять местами с предыдущим, получив массив  $[2, 3, 1]$ . Второй элемент является рекордом, поэтому его можно поменять местами с предыдущим, получив массив  $[3, 2, 1]$ . В этом массиве рекордов нет, поэтому к нему операцию из условия применить нельзя.

Таким образом, получаем, что в первом примере хорошими массивами являются  $[2, 1, 3]$ ,  $[2, 3, 1]$  и  $[3, 2, 1]$ . Посчитаем их стоимости:

- Массив префиксных сумм массива  $[2, 1, 3]$  равен  $[2, 3, 6]$ , поэтому его стоимость равна  $2 + 3 + 6 = 11$ .
- Массив префиксных сумм массива  $[2, 3, 1]$  равен  $[2, 5, 6]$ , поэтому его стоимость равна  $2 + 5 + 6 = 13$ .
- Массив префиксных сумм массива  $[3, 2, 1]$  равен  $[3, 5, 6]$ , поэтому его стоимость равна  $3 + 5 + 6 = 14$ .

Итого, получаем, что ответ на первый пример равен  $11 + 13 + 14 = 38$ .

## Задача Е. Прятки на дереве

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

Бельчата решили поиграть в прятки на дереве. Правила игры просты: выбирается вершина  $v$  — в ней будет находиться водящий бельчонок. Остальные бельчата разбегаются по дереву и прячутся в его вершинах. Чем дальше вершина от водящего, тем она безопаснее.

Изначально дерево состоит из единственной вершины с номером 1. Вам нужно обрабатывать запросы двух видов:

- $+ u$  — создание новой вершины (ее номер будет на единицу больше текущего размера дерева), которая присоединяется к вершине  $u$
- $? u$  — пусть водящий бельчонок находится в вершине  $u$ . Нужно вычислить расстояние до самой безопасной вершины дерева, а также количество таких вершин

Вершина  $u_i$  в запросе  $i$  вычисляется по формуле

$$u_i = (ans_{i-1} \cdot m + v_i - 1) \bmod sz + 1$$

где  $ans_{i-1}$  — сумма наибольшего расстояния и количества вершин из ответа на предыдущий запрос типа  $?$  (положим  $ans_0 = 0$ ),  $v_i$  — параметр запроса из ввода,  $sz$  — текущий размер дерева.

### Формат входных данных

В первой строке ввода находятся два целых числа  $q$  и  $m$  — количество запросов и флаг, отвечающий за генерацию запросов ( $1 \leq q \leq 300\,000$ ,  $m \in \{0, 1\}$ ).

В следующих  $q$  строках находятся сами запросы в соответствии с форматом, описанным в условии, по одному в строке. Гарантируется, что  $0 \leq v_i \leq q$ .

### Формат выходных данных

В ответ на каждый запрос второго типа выведите на отдельной строке два числа, разделенные пробелом — расстояние от вершины  $u_i$  до самой дальней от нее вершины дерева, а также количество вершин на таком расстоянии от  $u_i$ .

### Система оценки

В задаче 28 тестов (помимо тестов из условия), оценка потестовая. Тесты задачи можно разбить на следующие подгруппы:

Макс. Балл	$m$	Дополнительные Ограничения	Комментарий
0		—	тест из условия
9	$m = 0$	$q \leq 100$	
20	$m = 0$	все запросы типа + идут до всех запросов типа ?	
8	$m = 0$	диаметр итогового дерева не превосходит 50	
16	$m = 1$	в запросах типа ? $u_i = 1$ (после пересчета по формуле)	
20	$m = 1$	в запросах типа ? $u_i = 1$ или $u_i$ — сосед вершины 1 (после пересчета по формуле)	
15	$m = 0$	—	
12	$m = 1$	—	

### Примеры

стандартный ввод	стандартный вывод
10 0 + 1 ? 2 + 2 ? 2 + 3 ? 2 + 4 ? 1 + 4 ? 1	1 1 1 2 2 1 4 1 4 2
10 1 + 1 ? 1 + 1 + 0 + 0 + 1 + 1 ? 1 ? 5 ? 2	1 1 3 2 3 2 4 2

### Замечание

Во втором тесте из условия значения  $u_i$  (после пересчета по формуле) следующие:  
[1, 1, 1, 2, 2, 3, 3, 3, 3, 7].

Также приведем итоговое дерево из первого примера:

