

Задача А. Котёнок Гав

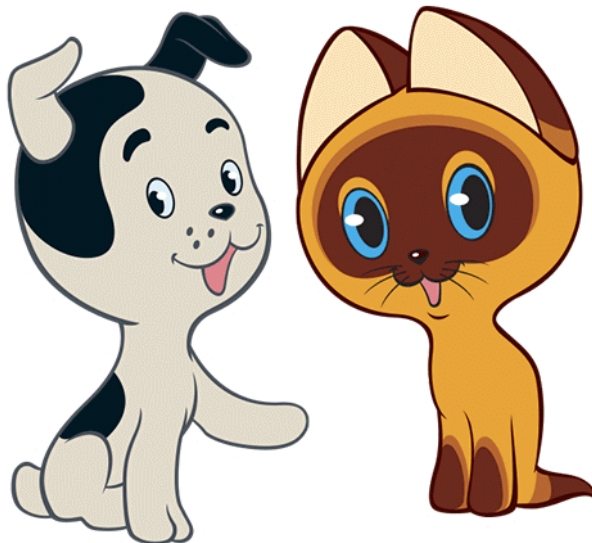
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Мультфильм «Котёнок по имени Гав» — экранизация одноимённой книги про приключения котёнка, которого зовут Гав, и его лучшего друга — щенка Шарика. Книга состоит из $c + d$ маленьких историй, где c — количество историй, главный герой которых — котёнок Гав, а d — количество историй, где главным героем является Шарик.

Студия решила создать мультсериал «Котёнок по имени Гав» из n выпусков, каждый из которых включает некоторое количество историй из книги. Чтобы каждый выпуск был интересным, были разработаны следующие требования:

1. Каждая история из книги попала в мультсериал ровно один раз;
2. В каждом выпуске должно быть хотя бы a историй про котёнка Гав;
3. В каждом выпуске должно быть хотя бы b историй про Шарика;
4. В каждом выпуске должно быть одинаковое количество историй.

По заданным n, a, b, c, d определите, можно ли экранизировать книгу, соблюдая все требования студии.



Шарик и Гав

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 10^{18}$) — запланированное количество выпусков мультсериала.

Во второй строке задано целое число a ($0 \leq a \leq 10^{18}$) — минимальное количество историй про котёнка Гав в каждом выпуске.

В третьей строке задано целое число b ($0 \leq b \leq 10^{18}$) — минимальное количество историй про щенка Шарика в каждом выпуске.

В четвертой строке задано целое число c ($0 \leq c \leq 2 \cdot 10^{18}$) — количество историй про котёнка Гав в книге.

В пятой строке задано целое число d ($0 \leq d \leq 2 \cdot 10^{18}$) — количество историй про щенка Шарика в книге.

Гарантируется, что $\max(c, d) > 0$.

Формат выходных данных

В единственной строке выходного файла выведите «Yes» (без кавычек, в любом регистре), если можно выполнить требования студии к экранизации и «No» (без кавычек, в любом регистре) — иначе.

Система оценки

Группа	Доп. ограничения	Баллы	Необх. группы
1	$n \leq 10^4, a \leq 10^4, b = 0, c \leq 2 \cdot 10^4, d = 0$	25	—
2	$n \leq 10^{18}, a \leq 10^{18}, b = 0, c \leq 2 \cdot 10^{18}, d = 0$	25	1
3	$n \leq 10^4, a \leq 10^4, b \leq 10^4, c \leq 2 \cdot 10^4, d \leq 2 \cdot 10^4$	25	1
4	—	25	1, 2, 3

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Примеры

стандартный ввод	стандартный вывод
3 1 1 4 5	Yes
4 3 3 10 20	No
3 1 0 5 0	No
3 1 0 6 0	Yes

Замечание

В первом примере истории можно распределить по выпускам следующим образом:

1. В первом выпуске 1 история про котёнка и 2 про щенка
2. Во втором выпуске 1 история про котёнка и 2 про щенка
3. В третьем выпуске 2 истории про котёнка и 1 про щенка

Таким образом, в каждом выпуске фигурирует хотя бы 1 история про щенка и хотя бы 1 история про котёнка, в сумме в выпусках 4 истории про котёнка и 5 историй про щенка.

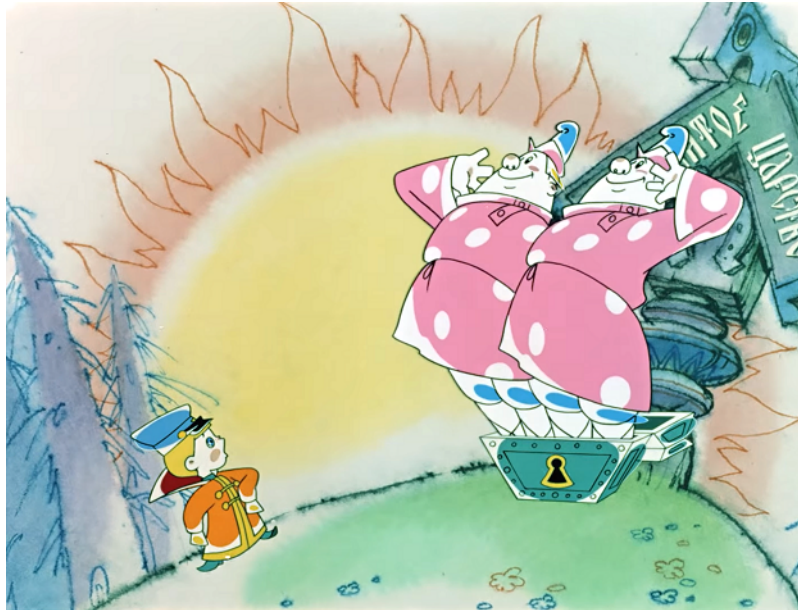
Во втором примере можно показать, что распределить истории по выпускам в связи с описанными в условии требованиями невозможно.

Задача В. Двое из ларца

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Запросы есть? Запросы есть!

— Народная поговорка



«А вы что, и конфеты за меня есть будете?!»

Царская жизнь — ну просто сказка. Хочешь — пирожное, хочешь — мороженное, только и делай, что ничего не делай! Вовка предпочитает царскую жизнь, поэтому его дорожка скатертью лежит в тридевятое царство. Там его ждут двое из ларца, одинаковых с лица. Говорят, что им не прикажешь — всё сделают, поэтому Вовка решил проверить их сноровку списком простых запросов.

Пусть есть последовательность a из n натуральных чисел. Двое из ларца отвечают на q запросов двух типов:

1. $upd(i, x)$ — заменить значение i -го элемента последовательности на число x .
2. $get(L, R)$ — для каждой подотрезки с границами в пределах от L до R найти сумму всех его элементов, затем найти сумму всех полученных значений. Формально¹:

$$get(L, R) = \sum_{l=L}^R \sum_{r=l}^R \sum_{i=l}^r a_i.$$

¹Знак $\sum_{i=a}^b i$. — математическое обозначение суммы, где: i — индекс суммирования, a — нижняя граница суммирования, b — верхняя граница суммирования.

Примеры:

$$\sum_{i=1}^5 i = 1 + 2 + 3 + 4 + 5 = 15$$

$$\sum_{i=1}^5 a_i = a_1 + a_2 + a_3 + a_4 + a_5$$

Вовка хочет знать, правильно ли двое из ларца выполняют его запросы и просит вас ему помочь (он бы проверил сам, но он не программист, и учиться ему не хочется).

Поскольку ответ на запрос второго типа может быть очень большим, в качестве ответа на запрос выведите остаток от деления на простое число $10^9 + 7$. Другими словами, если ответ на запрос равен val , то вам нужно вывести на экран число $val \bmod (10^9 + 7)$, где $a \bmod b$ обозначает остаток от деления числа a на число b .

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество элементов в последовательности a .

Во второй строке входного файла задана последовательность натуральных чисел $a_1, a_2, a_3, \dots, a_n$ ($0 \leq a_i \leq 10^9$). Все числа разделены пробелом.

В третьей строке входного файла задано целое число q ($1 \leq q \leq 2 \cdot 10^5$) — количество запросов.

В следующих q строках задано по три целых числа. Первое число t_i ($1 \leq t_i \leq 2$) в i -й строке — это тип i -го запроса.

Если $t_i = 1$, то описывается запрос первого типа, а во вводе два следующих числа — это i и x ($1 \leq i \leq n$), ($0 \leq x \leq 10^9$).

Если $t_i = 2$, то описывается запрос второго типа, а во вводе два следующих числа — это L и R ($1 \leq L \leq R \leq n$).

Формат выходных данных

Для каждого запроса второго вида выведите ответ по простому модулю $10^9 + 7$.

Система оценки

Группа	Доп. ограничения	Баллы	Необх. подгруппы
1	$n \leq 50, q \leq 100$, отсутствуют запросы первого типа	10	—
2	$n \leq 600, q \leq 600$, отсутствуют запросы первого типа	10	1
3	$n \leq 4 \cdot 10^3, q \leq 4 \cdot 10^3$, отсутствуют запросы первого типа	10	1, 2
4	отсутствуют запросы первого типа	20	1, 2, 3
5	$n \leq 50, q \leq 100$	10	1
6	$n \leq 600, q \leq 300$	10	1, 2, 5
7	$n \leq 4 \cdot 10^3, q \leq 4 \cdot 10^3$	10	1, 2, 3, 5, 6
8	—	20	1 — 7

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5 3 2 3 5 2 2 3 2 1 5	40 10 105
5 4 0 1 0 4 5 2 3 5 1 4 2 2 3 5 1 1 1 2 1 3	15 23 6
3 10000000000 10000000000 10000000000 1 2 1 3	999999937

Замечание

Разберём первые два запроса первого примера. В нём отсутствуют запросы обновления.

1. Ответ на первый запрос будет равен

$$(a_3) + (a_3 + a_4) + (a_3 + a_4 + a_5) + (a_4) + (a_4 + a_5) + (a_5) =$$

$$= (3) + (3 + 4) + (3 + 4 + 5) + (4) + (4 + 5) + (5) =$$

$$= (3) + (7) + (12) + (4) + (9) + (5) = 40.$$
2. Ответ на второй запрос будет равен

$$(a_2) + (a_2 + a_3) + (a_3) =$$

$$= (2) + (2 + 3) + (3) =$$

$$= (2) + (5) + (3) = 10.$$

Покажем промежуточные состояния последовательности во втором примере.

1. Изначально последовательность имеет вид $\{4, 0, 1, 0, 4\}$
2. После первого обновления последовательность имеет вид $\{4, 0, 1, 2, 4\}$
3. После второго обновления последовательность имеет вид $\{1, 0, 1, 2, 4\}$

В третьем примере искомая сумма единственного запроса равна 10^{10} . Однако необходимо вывести остаток этой суммы от деления на $10^9 + 7$, поэтому ответ равен 999999937.

Задача С. Дворец (типовой)

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



Программа по обмену премудростями

В своём приключении в Тридевятом Царстве Вовка повстречал Василис Премудрых, собравшихся на Слёт Юных Василис по обмену премудростями. «Премудрости — штука полезная» — решил Вовка, потому попросил Василис научить его некоторым премудростям. Каково же было его удивление, когда Василисы посадили его за парту и сказали решать задачи!

Одной из задач был расчёт количества стройматериалов для трёх n -этажных дворцов по типовому проекту.

Проект типового дворца задаётся следующими условиями:

- Основание каждого этажа является квадратом с **целыми** сторонами;
- Основание i -го этажа, начиная с самого верхнего, имеет площадь, не превосходящую i ;
- Площадь основания каждого этажа максимально возможная с учетом условий выше.



«Поработаем немного, рассчитаем за урок.»

Для расчёта количества стройматериалов Вовке было необходимо посчитать сумму периметров оснований всех этажей во всех трёх дворцах. К сожалению, учиться Вовка не хотел, и задачу так и не решил. А как у вас дела с премудростями?

Поскольку ответ может быть очень большим, вам необходимо вывести остаток от деления ответа на простое число $10^9 + 7$.

Формат входных данных

Во входных данных находятся несколько наборов входных данных.

В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют сами наборы входных данных.

В первой и единственной строке каждого набора задано одно целое число n ($1 \leq n \leq 10^{18}$) — число этажей в каждом из трёх дворцов.

Формат выходных данных

Для каждого набора входных данных выведите ответ на задачу по простому модулю $10^9 + 7$.

Система оценки

За Σn обозначим сумму n по всем наборам входных данных в тесте.

Группа	Доп. ограничения	Баллы	Необх. подгруппы
1	$\Sigma n \leq 10^7$	20	—
2	$\Sigma n \leq 10^{11}$	30	1
3	$\Sigma n \leq 10^{14}$	20	1, 2
4	—	30	1, 2, 3

Примеры из условия относятся к подгруппе 1.

Обратите внимание, что в третьей подгруппе нет дополнительных ограничений на Σn . Таким образом, Σn в третьей подгруппе может быть больше, чем 10^{18} .

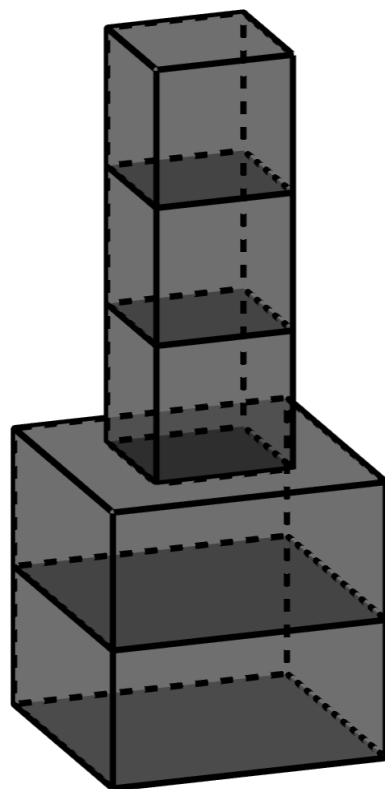
Пример

стандартный ввод	стандартный вывод
4	24
2	60
4	84
5	228
10	

Замечание

Рассмотрим третий пример, где $n = 5$.

- Площадь первого сверху этажа не должна превышать 1, откуда получаем, что длина стороны квадрата не превышает 1. Максимально возможная целая длина стороны — 1. Тогда периметр первого этажа равен $4 \cdot 1 = 4$.
- Площадь второго сверху этажа не должна превышать 2, откуда получаем, что длина стороны квадрата не превышает $\sqrt{2}$. Максимально возможная целая длина стороны — 1. Тогда периметр второго этажа равен $4 \cdot 1 = 4$.
- Площадь третьего сверху этажа не должна превышать 3, откуда получаем, что длина стороны квадрата не превышает $\sqrt{3}$. Максимально возможная целая длина стороны — 1. Тогда периметр третьего этажа равен $4 \cdot 1 = 4$.
- Площадь четвёртого сверху этажа не должна превышать 4, откуда получаем, что длина стороны квадрата не превышает 2. Максимально возможная целая длина стороны — 2. Тогда периметр четвёртого этажа равен $4 \cdot 2 = 8$.
- Площадь пятого сверху этажа не должна превышать 5, откуда получаем, что длина стороны квадрата не превышает $\sqrt{5}$. Максимально возможная целая длина стороны — 2. Тогда периметр пятого этажа равен $4 \cdot 2 = 8$.



$$i = 1, P = 1 \cdot 4 = 4$$

$$i = 2, P = 1 \cdot 4 = 4$$

$$i = 3, P = 1 \cdot 4 = 4$$

$$i = 4, P = 2 \cdot 4 = 8$$

$$i = 5, P = 2 \cdot 4 = 8$$

Суммируя, имеем 28. Так как дворца у нас три, то это число нужно умножить на 3. Тогда окончательный ответ равен $3 \cdot 28 = 84$.

Задача D. Опилки

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

Как все мы знаем, Винни Пух очень любит мёд. Но согласно медвежьим законам, мёд если есть, то его сразу нет. Вот Винни Пух и отправился к пчёлам за очередной порцией сладости.

По пути Винни Пух, как обычно, сочиняет песенки. Сегодня он хочет придумать ровно три песенки. Первая должна длиться не менее A секунд и быть придумана, когда Винни Пух поднимается в горку. Вторая песенка должна длиться не менее B секунд и быть придумана, когда Пух спускается с горы. А третья будет длиться не менее C секунд и придумываться, когда Винни снова поднимается. Чтобы у медведя не пропал аппетит (какой мёд без аппетита?), все песенки должны быть сочинены друг за другом без перерывов.

Если обозначить высоту Пуха в момент времени i как a_i , а общее время пути как n секунд, то подъем в горку означает, что на подотрезке для любого $i \neq n$ высота $a_i < a_{i+1}$, а спуск означает, что на подотрезке для любого $i \neq n$ высота $a_i > a_{i+1}$.

Винни Пух хочет знать, есть ли подходящий отрезок пути, где он сможет сочинить все три песенки и не потерять аппетит.

Винни Пуху сложно самому понять ответ, ведь у него в голове опилки (кричалки, вопилки, а также шумелки, пыхтелки и сопелки), поэтому он просит Вас ему помочь.



«Даже немножечко, чайная ложечка — это уже хорошо. А тем более целый горшок»

Формат входных данных

Первая строка содержит целое число n — количество секунд, которое Пух идет от дома до пчёл ($4 \leq n \leq 10^5$).

Во второй строке содержатся целые числа A , B и C — минимальная длительность трёх песенок, соответственно ($1 \leq A, B, C \leq n$).

В третьей строке содержится n целых чисел $a_1, a_2, a_3, \dots, a_n$, где a_i — высота, на которой будет находиться Пух в i -ю секунду ($-10^6 \leq a_i \leq 10^6$). Гарантируется, что любые два соседних значения высоты различны, т. е. $a_i \neq a_{i+1}$.

Формат выходных данных

Если подходящий отрезок времени существует, выведите два целых числа: время его начала и конца, в противном случае выведите «-1 -1» без кавычек.

Система оценки

Группа	Доп. ограничения	Баллы	Необх. подгруппы
1	$4 \leq n \leq 100$	20	—
2	$4 \leq n \leq 500$	30	1
3	$4 \leq n \leq 10^4$	30	1, 2
4	—	20	1, 2, 3

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Примеры

стандартный ввод	стандартный вывод
15 2 3 4 10 4 5 7 10 9 7 6 3 5 7 10 14 17 10	2 14
10 1 2 1 -10 0 10 15 7 9 11 3 4 9	-1 -1

Замечание

Пояснение к первому примеру.

На подотрезке $a_2, \dots, a_5$ Винни Пух поднимается в горку, первая песенка будет длиться 3 секунды. Далее на подотрезке $a_5, \dots, a_9$ он спускается и вторая песенка будет длиться 4 секунды. Наконец, на подотрезке $a_9, \dots, a_{14}$ Винни снова поднимается, третья песенка будет длиться 5 секунд.

Обратите внимание, что ответы «3 13», «3 14», «2 13» также являются верными.

Задача Е. Казнить нельзя помиловать

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

Запятые очень важны. Виктор Перестукин никогда не ставил запятые правильно и вообще не обращал на них никакого внимания. Но сейчас Витя попал в Страну невыученных уроков, и только правильно расставленные запятые могут спасти его жизнь!

У Вити есть предложение из n слов. Для каждого слова Витя знает уровень «хорошести» в виде целого числа. В предложение он может вписать от нуля до k запятых, и, если захочет, дописать перед запятой и/или после запятой слово «нельзя».

Витя хочет сделать уровень «хорошести» предложения наибольшим. Уровень «хорошести» предложения определяется как сумма «хорошести» обособленных частей предложения (обособленная часть предложения — это часть предложения, выделенная запятыми). «Хорошесть» обособленной части предложения — это сумма «хорошестей» её слов; эта сумма берется со знаком «+», если в блоке нет слова «нельзя», или со знаком «-», если «нельзя» есть.

Рассмотрим пример: «казнить помиловать». Пусть у слова «казнить» уровень «хорошести» равен «-100», а у слова «помиловать» — «100».

1. Если обойтись без запятых, «хорошесть» предложения будет равна 0: $(-100) + 100 = 0$
2. Если поставить запятую и добавить «нельзя» во вторую часть предложения, «хорошесть» предложения будет равна -200: «казнить, нельзя помиловать» = $(-100) + (-1) \cdot 100 = -200$
3. Если поставить запятую и добавить «нельзя» в первую часть предложения, «хорошесть» предложения будет равна 200: «казнить нельзя, помиловать» = $(-100) \cdot (-1) + 100 = 200$



«Куда ставить запятую?»

— Казнить нельзя, помиловать! — Воскликнул Витя.

С этой задачей он успешно справился (а ведь мог и голову потерять). Но это было лишь первое испытание. Помогите Вите ответить верно на все вопросы и отправиться домой. За вашу помощь Витя обещает всегда-всегда делать домашние задания, хорошо учиться, а ещё угостить вас вкусным тортом.

Обратите внимание, что запятые в начале или конце предложения быть не могут, в любой обособленной части предложения не может быть больше одного «нельзя».

Формат входных данных

В первой строке даны два целых числа n ($2 \leq n \leq 500$) и k ($1 \leq k \leq n - 1$) — количество слов в предложении и максимальное количество запятых, которое можно использовать в предложении, соответственно.

Во второй строке содержится n целых чисел $a_1, a_2, a_3, \dots, a_n$, где a_i ($-10^6 \leq a_i \leq 10^6$) — хорошесть i -го слова.

Формат выходных данных

Выведите единственное целое число — максимальный уровень «хорошесть», который можно получить в предложении.

Система оценки

Группа	Доп. ограничения	Баллы	Необх. подгруппы
1	$k = n - 1$ $0 \leq a_i \leq 10^4$	20	—
2	$k = n - 1$	20	1
3	$2 \leq n \leq 500$ $k = 1$	20	1, 2
4	$2 \leq n \leq 25$	20	—
5	—	20	1, 2, 3, 4

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Примеры

стандартный ввод	стандартный вывод
2 1 -100 100	200
8 3 2 -100 10 5 -10 3 -20 40	180

Замечание

Разберем второй пример.

Для достижения максимального ответа используется три запятых.

1. Первую запятую поставим после второго слова и перед запятой поставим «нельзя», получим блок « $(-1) \cdot (2 + (-100))$ ».
2. Вторую запятую поставим перед пятым словом, получим блок « $10 + 5$ ».
3. Третью запятую поставим после седьмого слова, перед третьей запятой добавим нельзя. Получим блоки « $((-10) + 3 + (-20)) \cdot (-1)$ » до запятой и «40» в конце предложения.

Тогда «хорошесть» предложения равна:

$$\begin{aligned} & \langle (2 + (-100)) \cdot (-1) \rangle + \langle 10 + 5 \rangle + \langle ((-10) + 3 + (-20)) \cdot (-1) \rangle + \langle 40 \rangle = \\ & = \langle 98 \rangle + \langle 15 \rangle + \langle 27 \rangle + \langle 40 \rangle = 180 \end{aligned}$$