

Задача А. Петя и прямоугольники

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Маленький Петя очень любит прямоугольники. Петя дал маме список прямоугольников, которые он хочет получить в подарок на Новый Год. Каждый прямоугольник характеризуется w и высотой h .

Мама хочет сделать Пете приятное и купить все прямоугольники из его списка. Мама отправилась в магазин и узнала, что цена одного прямоугольника равна его площади. К ее счастью, в магазине действует предновогодняя акция, позволяющая покупать прямоугольники не по одному, а сразу наборами. Стоимость одного набора равна ширине самого широкого прямоугольника, умноженной на высоту самого высокого прямоугольника из этого набора. Обратите внимание, что поворачивать прямоугольники (тем самым меняя местами ширину и высоту) нельзя. Помогите маме Пети купить все прямоугольники из списка ее сына, потратив на это наименьшее количество денег.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 2 \cdot 10^5$) — количество прямоугольников в списке Пети. В каждой из следующих n строк записаны по 2 целых положительных числа, не превышающих 10^6 , — ширина и высота очередного прямоугольника.

Формат выходных данных

Выведите одно число — наименьшее количество денег, которое может потратить мама чтобы купить Пете все прямоугольники из его списка.

Пример

стандартный ввод	стандартный вывод
4 100 1 15 15 20 5 1 100	500

Задача В. Снеговики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Зима. 2012 год. На фоне грядущего Апокалипсиса и конца света незамеченной прошла новость об очередном прорыве в областях клонирования и снеговиков: клонирования снеговиков. Вы конечно знаете, но мы вам напомним, что снеговик состоит из нуля или более вертикально поставленных друг на друга шаров, а клонирование — это процесс создания идентичной копии (клона).

В местечке Местячково учитель Андрей Сергеевич Учитель купил через интернет-магазин «Интернет-магазин аппаратов клонирования» аппарат для клонирования снеговиков. Теперь дети могут играть и даже играют во дворе в следующую игру. Время от времени один из них выбирает понравившегося снеговика, клонирует его и:

- либо добавляет ему сверху один шар;
- либо удаляет из него верхний шар (если снеговик не пустой).

Учитель Андрей Сергеевич Учитель записал последовательность действий и теперь хочет узнать суммарную массу всех построенных снеговиков.

Формат входных данных

Первая строка содержит количество действий n ($1 \leq n \leq 200\,000$). В строке номер $i + 1$ содержится описание действия i :

- $t\ m$ — клонировать снеговика номер t ($0 \leq t < i$) и добавить сверху шар массой m ($0 < m \leq 1000$);
- $t\ 0$ — клонировать снеговика номер t ($0 \leq t < i$) и удалить верхний шар. Гарантируется, что снеговик t не пустой.

В результате действия i , описанного в строке $i + 1$ создается снеговик номер i . Изначально имеется пустой снеговик с номером ноль.

Все числа во входном файле целые.

Формат выходных данных

Выведите суммарную массу построенных снеговиков.

Пример

стандартный ввод	стандартный вывод
8	74
0 1	
1 5	
2 4	
3 2	
4 3	
5 0	
6 6	
1 0	

Задача С. K -я порядковая статистика на отрезке

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

Дан массив из N неотрицательных чисел, строго меньших 10^9 . Вам необходимо ответить на несколько запросов о величине k -й порядковой статистики на отрезке $[l, r]$.

Формат входных данных

Первая строка содержит число N ($1 \leq N \leq 450\,000$) — размер массива.

Вторая строка может быть использована для генерации a_i — начальных значений элементов массива. Она содержит три числа a_1, l и m ($0 \leq a_1, l, m < 10^9$); для i от 2 до N

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9.$$

В частности, $0 \leq a_i < 10^9$.

Третья строка содержит одно целое число B ($1 \leq B \leq 1000$) — количество групп запросов.

Следующие B строк описывают одну группу запросов. Каждая группа запросов описывается 10 числами. Первое число G обозначает количество запросов в группе. Далее следуют числа x_1, l_x и m_x , затем y_1, l_y и m_y , затем, k_1, l_k и m_k ($1 \leq x_1 \leq y_1 \leq N$, $1 \leq k_1 \leq y_1 - x_1 + 1$, $0 \leq l_x, m_x, l_y, m_y, l_k, m_k < 10^9$). Эти числа используются для генерации вспомогательных последовательностей x_g и y_g , а также параметров запросов i_g, j_g и k_g ($1 \leq g \leq G$)

$$\begin{aligned} x_g &= ((i_{g-1} - 1) \cdot l_x + m_x) \bmod N + 1, & 2 \leq g \leq G \\ y_g &= ((j_{g-1} - 1) \cdot l_y + m_y) \bmod N + 1, & 2 \leq g \leq G \\ i_g &= \min(x_g, y_g), & 1 \leq g \leq G \\ j_g &= \max(x_g, y_g), & 1 \leq g \leq G \\ k_g &= (((k_{g-1} - 1) \cdot l_k + m_k) \bmod (j_g - i_g + 1)) + 1, & 2 \leq g \leq G \end{aligned}$$

Сгенерированные последовательности описывают запросы, g -й запрос состоит в поиске k_g -го по величине числа среди элементов отрезка $[i_g, j_g]$.

Суммарное количество запросов не превосходит 600 000.

Формат выходных данных

Выведите единственное число — сумму ответов на запросы.

Пример

стандартный ввод	стандартный вывод
5	15
1 1 1	
5	
1	
1 0 0 3 0 0 2 0 0	
1	
2 0 0 5 0 0 3 0 0	
1	
1 0 0 5 0 0 5 0 0	
1	
3 0 0 3 0 0 1 0 0	
1	
1 0 0 4 0 0 1 0 0	

Задача D. Соединение и разъединение

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

Вы когда-нибудь слышали про обход в глубину? Например, используя этот алгоритм, вы можете проверить является ли граф связным за время $O(E)$. Вы можете даже посчитать количество компонент связности за то же время.

А вы когда-нибудь слышали про систему непересекающихся множеств? Используя эту структуру, вы можете быстро обрабатывать запросы “Добавить ребро в граф” и “Посчитать количество компонент связности в графе”.

А вы когда-нибудь слышали о *динамической* задаче связности? В этой задаче вам необходимо обрабатывать три типа запросов:

1. Добавить ребро в граф.
2. Удалить ребро из графа.
3. Посчитать количество компонент связности в графе.

Можно считать, что граф является неориентированным. Изначально граф является пустым.

Формат входных данных

В первой строке находятся два целых числа N и K — количество вершин и количество запросов, соответственно ($1 \leq N \leq 300\,000$, $0 \leq K \leq 300\,000$). Следующие K строк содержат запросы, по одному в строке. Каждый запрос имеет один из трех типов:

1. $+ u v$: Добавить ребро между вершинами u и v . Гарантируется, что такого ребра нет.
2. $- u v$: Удалить ребро между u и v . Гарантируется, что такое ребро есть.
3. $?$: Посчитать количество компонент связности в графе.

Вершины пронумерованы целыми числами от 1 до N . Во всех запросах $u \neq v$.

Формат выходных данных

Для каждого запроса типа ‘?’, Выведите количество компонент связности в момент запроса.

Пример

стандартный ввод	стандартный вывод
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

Задача Е. Комбокамень

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

Одна большая компания разрабатывает компьютерную игру «Комбокамень», которая должна мигом перевернуть всю индустрию. Правила игры достаточно сложные, и на реализацию серверного движка, моделирующего ход игры, был объявлен конкурс. От вас требуется реализовать подобный серверный движок.

Суть игры — игроки умеют призывать на арену и усиливать существ, используя заклинания, а также заставлять их сражаться друг с другом. Каждое существо имеет два параметра — численное значение атаки a и численное значение оставшегося здоровья h . Для краткости будем обозначать параметры существа как (a, h) . Исходно на арене нет существ.

Игроку доступны следующие заклинания:

- *Призыв существа*: Призвать новое существо с характеристиками $(1, 1)$. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Благословение силы*: Удвоить атаку выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h) , то после этого действия оно будет иметь характеристики $(2a, h)$.
- *Божественный дух*: Удвоить здоровье выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h) , то после этого действия оно будет иметь характеристики $(a, 2h)$.
- *Копия из лавы*: Призвать новое существо, которое будет иметь такие же характеристики, как и выбранное заклинанием существо. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Сражайся!*: Заставить двух различных существ сразиться. Во время сражения оба существа одновременно наносят друг другу по одному удару, уменьшая количество здоровья соперника на значение своей атаки. Так, если сражаются два существа с характеристиками (a_1, h_1) и (a_2, h_2) , то после сражения они будут иметь характеристики $(a_1, h_1 - a_2)$ и $(a_2, h_2 - a_1)$, соответственно. Если после сражения у существа остается 0 или меньше единиц здоровья, оно умирает и больше не может участвовать в игре.

От серверного движка, на реализацию которого объявлен конкурс, требуется способность промоделировать все события и для каждого созданного во время игры существа вывести номер хода, на котором оно погибло, либо определить, что оно осталось живо к концу игры.

Кроме того, движок должен корректно обрабатывать случаи, когда игрок пытается взаимодействовать с мертвыми по мнению сервера существами: если заклинание *Благословение силы*, *Божественный дух* или *Сражайся!* обращено к уже мертвому существу, то не должно произойти ничего. Если заклинание *Копия из лавы* применено к мертвому существу, создается его мертвая копия с такими же характеристиками, но умершая на текущем ходу, в момент копирования.

Формат входных данных

В первой строке дано число n — количество совершенных ходов ($1 \leq n \leq 250\,000$).

В следующих n строках даны ходы, пришедшие к серверному движку, в следующем формате:

- 1 — применить заклинание *Призыв существа*;
- 2 i — применить заклинание *Благословение силы* к существу с номером i ;
- 3 i — применить заклинание *Божественный дух* к существу с номером i ;
- 4 i — применить заклинание *Копия из лавы* к существу с номером i ;
- 5 $i j$ — применить заклинание *Сражайся!* к существам с номерами i и j .

Гарантируется, что любые упомянутые в запросах существа к моменту запроса уже были призваны, но, возможно, могут уже быть мертвы.

Формат выходных данных

В первой строке выведите одно целое число k — количество существ, призванных за время игры.

В следующей строке выведите k целых чисел t_1, t_2, \dots, t_k — если существо с номером i осталось живо к концу игры, то t_i должно быть равно -1 , иначе t_i должно быть равно номеру хода, на котором оно погибло.

Пример

стандартный ввод	стандартный вывод
16	5
1	13 5 14 -1 16
2 1	
3 1	
1	
5 1 2	
3 1	
1	
3 3	
3 3	
4 1	
5 1 3	
3 3	
5 1 3	
5 4 3	
5 4 3	
4 1	

Замечание

В таблице можно увидеть, как изменялись характеристики существ в первом примере.

ход	1	2	3	4	5
0	-	-	-	-	-
1	(1, 1)	-	-	-	-
2	(2, 1)	-	-	-	-
3	(2, 2)	-	-	-	-
4	(2, 2)	(1, 1)	-	-	-
5	(2, 1)	мертво	-	-	-
6	(2, 2)	мертво	-	-	-
7	(2, 2)	мертво	(1, 1)	-	-
8	(2, 2)	мертво	(1, 2)	-	-
9	(2, 2)	мертво	(1, 4)	-	-
10	(2, 2)	мертво	(1, 4)	(2, 2)	-
11	(2, 1)	мертво	(1, 2)	(2, 2)	-
12	(2, 1)	мертво	(1, 4)	(2, 2)	-
13	мертво	мертво	(1, 2)	(2, 2)	-
14	мертво	мертво	мертво	(2, 1)	-
15	мертво	мертво	мертво	(2, 1)	-
16	мертво	мертво	мертво	(2, 1)	мертво

Задача F. Минимизируй!

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

Дан массив целых чисел a длины n . Поступает q запросов двух типов:

- 1 $l r x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $\min(a_i, x)$.
- 2 $l r$. Необходимо вывести сумму элементов массива a на отрезке от l до r включительно.

Формат входных данных

В первой строке входных данных дано целое число n ($1 \leq n \leq 300\,000$) — количество элементов массива a .

Во второй строке даны n целых чисел a_i ($1 \leq a_i \leq 10^9$) — элементы массива a .

В третьей строке дано целое число q ($1 \leq q \leq 300\,000$) — количество запросов.

В последующих q строках даны запросы по одному в строке.

Запрос первого типа задается так: 1 $l r x$

Где $1 \leq l \leq r \leq n$, $1 \leq x \leq 10^9$ — целые числа. Это означает, что все элементы массива a на отрезке от l до r нужно заменить на минимум из текущего значения и x .

Запрос второго типа задается так: 2 $l r$

Где $1 \leq l \leq r \leq n$ — целые числа. Это означает, что нужно вывести сумму элементов массива a на отрезке от l до r .

Формат выходных данных

Для каждого запроса второго типа выведите в отдельной строке сумму элементов на соответствующем отрезке.

Примеры

стандартный ввод	стандартный вывод
3	7
1 4 2	6
5	3
2 1 3	
1 1 3 3	
2 1 3	
1 1 3 1	
2 1 3	
7	118
1 7 2 4 8 4 100	117
7	9
1 3 6 3	17
2 2 7	
1 2 3 5	
2 1 7	
1 1 7 3	
2 1 4	
2 2 7	

Задача G. Минимизируй, прибавляй!

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

Дан массив целых чисел a длины n . Поступает q запросов трех типов:

- $1\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $\min(a_i, x)$.
- $2\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $a_i + x$.
- $3\ l\ r$. Необходимо вывести сумму элементов массива a на отрезке от l до r включительно.

Формат входных данных

В первой строке входных данных дано целое число n ($1 \leq n \leq 300\,000$) — количество элементов массива a .

Во второй строке даны n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — элементы массива a .

В третьей строке дано целое число q ($1 \leq q \leq 300\,000$) — количество запросов.

В последующих q строках даны запросы по одному в строке.

Запрос первого типа задается так: $1\ l\ r\ x$

Где $1 \leq l \leq r \leq n$, $-10^9 \leq x \leq 10^9$. Это означает, что все элементы массива a на отрезке от l до r нужно заменить на минимум из текущего значения и x .

Запрос второго типа задается так: $2\ l\ r\ x$

Где $1 \leq l \leq r \leq n$, $-10^7 \leq x \leq 10^7$. Это означает, что ко всем элементам массива a на отрезке от l до r нужно прибавить x .

Запрос третьего типа задается так: $3\ l\ r$

Где $1 \leq l \leq r \leq n$. Это означает, что нужно вывести сумму элементов массива a на отрезке от l до r .

Формат выходных данных

Для каждого запроса третьего типа выведите в отдельной строке сумму элементов на соответствующем отрезке.

Примеры

стандартный ввод	стандартный вывод
3 1 4 2 9 3 1 3 1 1 3 3 3 1 3 1 1 3 1 3 1 3 2 1 3 5 3 1 3 1 1 3 3 3 1 3	7 6 3 18 9
7 1 7 2 4 8 4 100 10 1 3 6 3 3 2 7 1 2 3 5 2 3 4 -10 3 1 7 1 1 7 3 3 1 4 3 2 7 2 1 7 5 3 1 7	118 97 -11 -3 33

Задача Н. Минимизируй, прибавляй, НОДируй и их друзья

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Дан массив целых чисел a длины n . Поступает q запросов восьми типов:

- $1\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $\min(a_i, x)$.
- $2\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $\max(a_i, x)$.
- $3\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на x .
- $4\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $a_i + x$.
- $5\ l\ r$. Необходимо вывести сумму элементов массива a на отрезке от l до r включительно.
- $6\ l\ r$. Необходимо вывести минимум элементов массива a на отрезке от l до r включительно.
- $7\ l\ r$. Необходимо вывести максимум элементов массива a на отрезке от l до r включительно.
- $8\ l\ r$. Необходимо вывести НОД (наибольший общий делитель) элементов массива a на отрезке от l до r включительно.

Формат входных данных

В первой строке входных данных дано целое число n ($1 \leq n \leq 200\,000$) — количество элементов массива a .

Во второй строке даны n целых чисел a_i ($1 \leq a_i \leq 10^9$) — элементы массива a .

В третьей строке дано целое число q ($1 \leq q \leq 200\,000$) — количество запросов.

В последующих q строках даны запросы по одному в строке.

Запрос первого типа задается так: $1\ l\ r\ x$

Где $1 \leq l \leq r \leq n$, $1 \leq x \leq 10^9$ — целые числа. Это означает, что все элементы массива a на отрезке от l до r нужно заменить на минимум из текущего значения и x .

Запрос второго типа задается так: $2\ l\ r\ x$

Где $1 \leq l \leq r \leq n$, $1 \leq x \leq 10^9$ — целые числа. Это означает, что все элементы массива a на отрезке от l до r нужно заменить на максимум из текущего значения и x .

Запрос третьего типа задается так: $3\ l\ r\ x$

Где $1 \leq l \leq r \leq n$, $1 \leq x \leq 10^9$ — целые числа. Это означает, что все элементы массива a на отрезке от l до r нужно заменить на x .

Запрос четвертого типа задается так: $4\ l\ r\ x$

Где $1 \leq l \leq r \leq n$, $1 \leq x \leq 10^7$ — целые числа. Это означает, что ко всем элементам массива a на отрезке от l до r нужно прибавить x .

Запрос пятого типа задается так: $5\ l\ r$

Где $1 \leq l \leq r \leq n$ — целые числа. Это означает, что нужно вывести сумму элементов массива a на отрезке от l до r .

Запрос шестого типа задается так: $6\ l\ r$

Где $1 \leq l \leq r \leq n$ — целые числа. Это означает, что нужно вывести минимум элементов массива a на отрезке от l до r .

Запрос седьмого типа задается так: $7\ l\ r$

Где $1 \leq l \leq r \leq n$ — целые числа. Это означает, что нужно вывести максимум элементов массива a на отрезке от l до r .

Запрос восьмого типа задается так: $8\ l\ r$

Где $1 \leq l \leq r \leq n$ — целые числа. Это означает, что нужно вывести наибольший общий делитель элементов массива a на отрезке от l до r .

Формат выходных данных

Для каждого запроса 5, 6, 7 и 8 типов выведите в отдельной строке ответ.

Пример

стандартный ввод	стандартный вывод
7	71
1 2 3 4 5 6 7	1
20	16
4 2 7 10	1
5 1 6	90
6 1 6	14
7 1 6	17
8 1 6	1
2 1 6 14	74
5 2 7	12
6 2 7	14
7 2 7	2
8 2 7	101
1 2 7 12	12
5 1 6	15
6 1 6	1
7 1 6	
8 1 6	
3 2 6 15	
5 1 7	
6 1 7	
7 1 7	
8 1 7	

Задача I. Контрол Икс Контрол Вэ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	960 мегабайт

Эта задача была нагло скопипасчена преподавателями со стороннего ресурса.

Вам дана строка S состоящая из цифр 1, 2, 3. Длину строки обозначим за $|S|$.

У вас есть курсор. Будем обозначать его позицию за ℓ . Означать позиция будет следующее:

- Если $\ell = 0$, то курсор находится перед первым символом S .
- Если $\ell = |S|$, то курсор находится после последнего символа S .
- Если $0 < \ell < |S|$, то курсор находится между S_ℓ и $S_{\ell+1}$.

Будем называть S_{left} префикс строки S до позиции курсора, и S_{right} суффикс строки S после курсора. Иначе говоря, курсор «разрезает» строку S на S_{left} и S_{right} .

Еще у нас есть строка C , изначально пустая. С ней доступны операции трех видов:

- **Вырезать.** Присваивает $C \leftarrow S_{\text{right}}$, затем присваивает $S \leftarrow S_{\text{left}}$.
- **Вставить.** Приписывает строку C к строке S справа.
- **Подвинуть.** Сдвигает курсор на 1 вправо, увеличивает ℓ на единицу.

Изначально курсор находится в позиции $\ell = 0$. Затем мы делаем следующий алгоритм:

1. **Подвинуть.**
2. **Вырезать.**
3. **Вставить**, выполняется S_ℓ раз.
4. Если $\ell = |S|$, остановить работу. Иначе начать сначала с шага 1.

Собственно, что от вас требуется:

Вам дается строка S , и два натуральных числа x и n . Выполним наш алгоритм и дождемся момента, когда ℓ впервые достигнет значения x ; это произойдет после очередного **Подвинуть**. Чему будут равны последние n символов у S_{left} в этот момент?

Боги копиясты утверждают, что в ходе работы алгоритма ℓ достигнет значения x .

Формат входных данных

В первой строке вводится число t — число тестов. Тесты перечисляются один за другим.

В первой строке для каждого теста вводятся числа x и n . В следующей строке для этого теста будет введена изначально строка S .

- $1 \leq t \leq 250$
- $1 \leq |S| \leq 200$
- $n \leq x \leq 10^{16}$
- $1 \leq n \leq 200$

Формат выходных данных

Для каждого теста, выведите в новой строке n символов — ответ на тест.

Пример

стандартный ввод	стандартный вывод
3	323
7 3	333333333
2323	31332133213
20 10	
333	
24 11	
132133	

Замечание

Разберем третий тест. Изначально у нас есть $S = 132133$, $\ell = 0$ и $C = \varepsilon$ (пустая строка). Потом происходит следующее:

- Step 1, *Подвинуть*: $\ell = 1$.
- Step 2, *Вырезать*: $S = 1$ и $C = 32133$.
- Step 3, *Вставить* $S_\ell = 1$ раз: получили $S = 132133$.
- Step 4: $\ell = 1 \neq |S| = 6$, начнем заново.
- Step 1, *Подвинуть*: $\ell = 2$.
- Step 2, *Вырезать*: $S = 13$ и $C = 2133$.
- Step 3, *Вставить* $S_\ell = 3$ раза: $S = 13213321332133$.
- Step 4: $\ell = 2 \neq |S| = 14$, начнем сначала.
- Step 1, *Подвинуть*: $\ell = 3$.
- Step 2, *Вырезать*: $S = 132$ и $C = 13321332133$.
- Step 3, *Вставить* $S_\ell = 2$ раза: $S = 1321332133213313321332133$.
- Step 4: $\ell = 3 \neq |S| = 25$, начнем сначала.
- Step 1, *Подвинуть*: $\ell = 4$.
- Step 2, *Вырезать*: $S = 1321$ и $C = 332133213313321332133$.
- Step 3, *Вставить* $S_\ell = 1$ раз: $S = 1321332133213313321332133$.
- Step 4: $\ell = 4 \neq |S| = 25$, Начнем сначала.
- Step 1, *Подвинуть*: $\ell = 5$.
- И так далее ...

В какой-то момент, ℓ достигнет $x = 24$, и S будет какой-то очень длинной строкой. Можно проверить, что в этот момент времени последние $n = 11$ символов S слева от курсора будут 31332133213.

Задача J. Урны и шары

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Пусть у вас есть n урн, в каждой из которых лежит по одному шару. Урна с номером i содержит шарик под номером i . У вас есть специальное устройство, которое позволяет перемещать шарики. Им чрезвычайно просто пользоваться: сначала вы выбираете некоторый отрезок последовательных урн. После этого вы выбираете некоторый другой отрезок последовательных урн такой же длины, как и исходный, и затем шарики из урн первого отрезка перемещаются в соответствующие урны второго отрезка.

Дана последовательность перемещений. Установите, в какой урне окажется каждый шарик.

Формат входных данных

Первая строка входных данных содержит два числа n и m — число урн и число перемещений, соответственно ($1 \leq n \leq 100\,000$, $1 \leq m \leq 50\,000$). Каждая из следующих m строк содержит три числа $count_i$, $from_i$ и to_i , которые означают одновременное перемещение всех шариков из урны $from_i$ в урну to_i , всех шариков из урны $from_i + 1$ в урну $to_i + 1$, ..., всех шариков из урны $from_i + count_i - 1$ в урну $to_i + count_i - 1$ ($1 \leq count_i, from_i, to_i \leq n$, $\max(from_i, to_i) + count_i \leq n + 1$).

Формат выходных данных

Выведите n чисел — итоговые позиции каждого шарика.

Пример

стандартный ввод	стандартный вывод
2 3	1 1
1 1 2	
1 2 1	
1 2 1	

Задача К. Запросы на арифметические прогрессии

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

Дан массив целых чисел s длины n и q запросов трех типов:

1. Даны l, r, a и b . Нужно для всех i на отрезке от l до r прибавить $a + (i - l) \cdot b$ к s_i .
2. Даны l, r, a и b . Нужно для всех i на отрезке от l до r присвоить элементу s_i значение $a + (i - l) \cdot b$.
3. Даны l и r . Нужно найти длину наибольшего подотрезка отрезка $[l, r]$, элементы которого образуют арифметическую прогрессию.

Формат входных данных

В первой строке входных данных дано два целых положительных числа n и q ($1 \leq n, q \leq 300\,000$).

Во второй строке входных данных дано n целых чисел s_i ($-10^6 \leq s_i \leq 10^6$) — элементы массива s .

В последующих q строках даны запросы по одному в строке.

Запрос первого типа задается так: $1\ l\ r\ a\ b$

Где $1 \leq l \leq r \leq n$, $-10^6 \leq a, b \leq 10^6$ — целые числа. Это означает, что ко всем элементам массива s на отрезке от l до r нужно прибавить $a + (i - l) \cdot b$.

Запрос второго типа задается так: $2\ l\ r\ a\ b$

Где $1 \leq l \leq r \leq n$, $-10^6 \leq a, b \leq 10^6$ — целые числа. Это означает, что всем элементам массива s на отрезке от l до r нужно присвоить $a + (i - l) \cdot b$.

Запрос третьего типа задается так: $3\ l\ r$

Где $1 \leq l \leq r \leq n$ — целые числа. В ответ на этот запрос нужно вывести максимальную длину подотрезка отрезка $[l, r]$ массива s , обращующего арифметическую прогрессию.

Формат выходных данных

Для каждого запроса третьего типа необходимо вывести ответ в отдельной строке.

Примеры

стандартный ввод	стандартный вывод
10 6 1 2 3 4 1 2 3 4 5 5 3 1 10 1 1 4 -1 -1 3 1 10 3 9 10 2 5 10 -2 -2 3 1 10	5 6 2 7
10 5 1 2 3 4 1 2 3 4 5 5 3 1 10 1 1 10 1 2 3 1 10 2 1 10 3 4 3 1 10	5 5 10
10 5 1 2 3 4 1 2 3 4 5 5 3 1 4 3 4 5 3 2 4 3 5 9 3 10 10	4 2 3 5 1
10 5 1 2 3 4 1 2 3 4 5 5 2 10 10 6 1 3 5 10 1 5 5 4 1 3 1 5 3 1 6	6 5 5
10 5 1 2 3 4 1 2 3 4 5 5 1 1 4 -1 -1 3 1 5 3 4 5 1 5 10 -1 -1 3 1 10	4 2 9

Задача L. Дели, прибавляй!

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

Дан массив целых чисел a длины n . Поступает q запросов четырех типов:

- $1\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $a_i + x$.
- $2\ l\ r\ x$. Для каждого i на отрезке от l до r включительно нужно заменить a_i на $\lfloor \frac{a_i}{x} \rfloor$ ($\lfloor \cdot \rfloor$ — это округление вниз).
- $3\ l\ r$. Необходимо вывести минимум элементов массива a на отрезке от l до r включительно.
- $4\ l\ r$. Необходимо вывести сумму элементов массива a на отрезке от l до r включительно.

Формат входных данных

В первой строке входных данных даны два целых числа n и q ($1 \leq n, q \leq 200\,000$) — количество элементов массива a и количество запросов.

Во второй строке даны n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — элементы массива a .

В последующих q строках даны запросы по одному в строке.

Запрос первого типа задается так: $1\ l\ r\ x$

Где $0 \leq l \leq r < n$, $-10^7 \leq x \leq 10^7$ — целые числа. Это означает, что ко всем элементам массива a на отрезке от l до r нужно прибавить x .

Запрос второго типа задается так: $2\ l\ r\ x$

Где $0 \leq l \leq r < n$, $1 \leq x \leq 10^9$ — целые числа. Это означает, что все элементы массива a на отрезке от l до r нужно поделить на x и округлить вниз.

Запрос третьего типа задается так: $3\ l\ r$

Где $0 \leq l \leq r < n$ — целые числа. Это означает, что нужно вывести минимум элементов массива a на отрезке от l до r .

Запрос четвертого типа задается так: $4\ l\ r$

Где $0 \leq l \leq r < n$ — целые числа. Это означает, что нужно вывести сумму элементов массива a на отрезке от l до r .

Формат выходных данных

Для каждого запроса 3 и 4 типов выведите в отдельной строке ответ.

Пример

стандартный ввод	стандартный вывод
7 7	-10
10 -6 4 7 12 1 0	-10
2 1 4 4	-34
1 0 5 -8	-36
3 0 6	-26
3 1 5	
4 0 6	
4 1 5	
4 2 6	

Замечание

Обратите внимание на то, что элементы массива нумеруются с нуля.

Задача М. Исторический максимум

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

Дан массив a . Поступают запросы прибавления на отрезке и поиска максимума исторических максимумов на отрезке. Необходимо их обрабатывать.

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 300\,000$) — количество элементов массива a .

Во второй строке даны n чисел — элементы массива a ($-10^9 \leq a_i \leq 10^9$).

В третьей строке дано число q ($1 \leq q \leq 300\,000$) — количество запросов.

В последующих q строках даны запросы.

Запрос прибавления на отрезке задается следующим образом: 1 ql qr x

Это означает, что на отрезке от ql до qr включительно ($1 \leq ql \leq qr \leq n$) нужно прибавить ко всем числам x ($-10^9 \leq x \leq 10^9$).

Запрос поиска исторического максимума задается следующим образом: 2 ql qr

Это означает, что нужно найти максимум исторических максимумов на отрезке от ql до qr включительно ($1 \leq ql \leq qr \leq n$).

Формат выходных данных

Для каждого запроса второго типа выведите ответ в отдельной строке.

Пример

стандартный ввод	стандартный вывод
5	6
1 2 3 4 5	6
5	6
1 1 4 2	
2 2 5	
1 3 5 -5	
2 4 5	
2 1 5	

Задача N. River Land

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Есть страна с большой речной системой. Города в этой стране пронумерованы от 1 до N ; город № 1 является столицей. Речная система образует корневое дерево, так что столица является корнем. Каждая река соединяет два города и течет к корню. Для каждого некоренного города i есть река, которая течет из этого города в город p_i . В каждом городе также есть порт.

В стране есть только два способа путешествовать между городами: на лодке и на пароме. Лодка может двигаться только вниз по течению (то есть к столице), в то время как паром может двигаться в обоих направлениях. Есть прямые паромные маршруты из каждого порта в любой другой порт.

Изначально все порты открыты. Иногда некоторые порты могут закрываться или открываться снова (возможно, несколько раз). Порт в столице всегда открыт. Не разрешается путешествовать на пароме из закрытого порта, но допускается путешествие на пароме из открытого порта в закрытый порт. Путешествие на лодке всегда возможно, даже из закрытого порта.

Иногда вы хотите отправиться в путешествие. Вы уже выбрали город, в котором начнете путешествие. Однако у вас достаточно денег только для одной поездки на пароме. Поэтому вы хотите совершить путешествие следующим образом:

- Выбрать город b и отправиться в этот город на лодке. Должна быть возможность путешествовать от v до b на лодке (в частности, $v = b$ разрешено).
- Решить либо продолжить движение из города b на пароме, либо завершить поездку в этом городе.
- Если вы решаете продолжить путь на пароме, вы выбираете город f и отправляетесь на пароме от b до f . Также должна быть возможность путешествовать от b до f на пароме: не разрешается путешествовать вдоль любой реки дважды (независимо от направления), т.е., когда вы решаете продолжить путь на пароме, кратчайшие пути между (v, b) и (b, f) не должны иметь общих рек.
- Вы любите путешествовать на пароме, поэтому чем больше рек вы посещаете во время путешествия на пароме, тем счастливее вы становитесь. Также в каждом городе есть фиксированная красота; давайте обозначим красоту города i как a_i . Затем, для поездки со значением T (для каждой поездки задано своё значение) определим, что вы получаете счастье $a_b + D \cdot T$, где D — длина маршрута парома (количество рек, посещаемых во время путешествия на пароме; возможно, ноль, если продолжить путешествие на пароме из города b невозможно).

Вы должны обработать Q запросов. Есть три типа запросов:

1. $- v$: порт в городе v закрывается. Гарантируется, что порт в городе v был открыт до этого запроса.
2. $+ v$: порт в городе v вновь открывается. Гарантируется, что порт в городе v был закрыт до этого запроса.
3. $? v T$: вы хотите совершить поездку со значением T из города v .

Для каждого запроса третьего типа найдите максимальное счастье, которое вы можете получить от такой поездки.

Формат входных данных

Первая строка содержит два целых числа N и Q ($2 \leq N \leq 3 \cdot 10^5$, $1 \leq Q \leq 3 \cdot 10^5$).

Вторая строка содержит $N - 1$ целое число p_2, p_3, \dots, p_N ($1 \leq p_i < i$).

Третья строка содержит N целых чисел a_1, a_2, \dots, a_N ($1 \leq a_i \leq 10^9$).

Следующие Q строк описывают запросы в формате, описанном в условии. $1 \leq v \leq N$, $1 \leq T \leq 10^9$.

Формат выходных данных

Для каждого запроса третьего типа выведите ответ на него в отдельной строке.

Система оценки

Подзадача	Доп. ограничения	Баллы
1	$N, Q \leq 1000$	30
2	$N, Q \leq 10^5$	35
3	—	35

Пример

стандартный ввод	стандартный вывод
10 9	57
1 2 3 2 2 6 3 8 8	42
30 20 6 13 8 40 7 9 13 1	33
? 4 11	30
- 4	
? 4 11	
- 7	
? 10 6	
+ 7	
- 6	
- 2	
? 7 4	

Замечание

В первом запросе вы путешествуете на пароме из города 4 в город 7.

Во втором запросе вы не можете путешествовать на пароме из города 4, поэтому вы должны использовать второй по выгодности маршрут, который идет от города 2 к городу 7.

В третьем запросе вы едете в город 8 на лодке, а затем в город 7 на пароме.

В четвертом запросе вы едете в столицу (город № 1) на лодке и не путешествуете на пароме.

Задача О. Персистентная очередь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Реализуйте персистентную очередь.

Формат входных данных

Первая строка содержит количество действий n ($1 \leq n \leq 200\,000$). В строке номер $i + 1$ содержится описание действия i :

- $1\ t\ m$ — добавить в конец очереди номер t ($0 \leq t < i$) число m ;
- $-1\ t$ — удалить из очереди номер t ($0 \leq t < i$) первый элемент.

В результате действия i , описанного в строке $i + 1$ создается очередь номер i . Изначально имеется пустая очередь с номером ноль.

Все числа во входном файле целые, и помещаются в знаковый 32-битный тип.

Формат выходных данных

Для каждой операции удаления выведите удаленный элемент на отдельной строке.

Пример

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	