

## Задача А. Параллелепипед

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

Мальчик Петя очень любит коллекционировать прямоугольники. Сегодня он достал свою коллекцию и стал складывать прямоугольные параллелепипеды из прямоугольников.

Пете стало интересно, какой максимальный объём может иметь параллелепипед, сложенный из имеющихся у него прямоугольников. Прямоугольники нельзя сгибать, разрезать, также ни один из них не должен выходить за края параллелепипеда; но прямоугольники можно поворачивать (таким образом, прямоугольники  $(1, 2)$  и  $(2, 1)$  считаются одинаковыми).

### Формат входных данных

В первой строке входного файла даётся одно число  $n$  ( $6 \leq n \leq 200\,000$ ) — количество прямоугольников в коллекции Пети.

Каждая из следующих строк содержит пару целых чисел — длину и ширину очередного прямоугольника.

### Формат выходных данных

В единственной строке выведите единственное число: наибольший объём параллелепипеда, который можно сложить из имеющихся у Пети прямоугольников. Если ни один параллелепипед из данных прямоугольников составить нельзя, выведите единственное число  $-1$ .

### Примеры

| стандартный ввод                            | стандартный вывод |
|---------------------------------------------|-------------------|
| 6<br>3 6<br>6 9<br>9 3<br>6 3<br>3 9<br>9 6 | 162               |
| 6<br>1 1<br>1 1<br>1 1<br>1 1<br>1 1<br>1 1 | 1                 |
| 6<br>1 2<br>2 3<br>3 4<br>4 5<br>5 6<br>6 1 | -1                |

## Задача В. Посредник

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

Вам дан граф на  $n$  вершинах, в котором изначально нет рёбер. Далее приходят запросы двух типов. Первый тип задается как  $1\ u\ v$ , где  $u$  и  $v$  – вершины, и требует провести ребро между двумя данными вершинами. Гарантируется, что  $u$  и  $v$  до запроса находились в разных компонентах связности. Второй тип:  $2\ u\ v$ , где  $u, v$  снова вершины. Требуется вывести номер вершины  $w$ , которая соединена ребром и с  $u$ , и с  $v$ . Если такой  $w$  не существует, то выведите 0.

Запросы даны в зашифрованном виде. Пусть  $x_i$  – ответ на  $i$ -й запрос 2-го типа, также положим  $x_0 = 0$ . Тогда, чтобы восстановить тройку чисел  $A, B, C$  для запроса, по данным вместо них  $a, b, c$ , нужно сделать следующее:

$$A = 1 + (((a \cdot (1 + x_l)) \bmod 998244353) \bmod 2)$$

$$B = 1 + (((b \cdot (1 + x_l)) \bmod 998244353) \bmod n)$$

$$C = 1 + (((c \cdot (1 + x_l)) \bmod 998244353) \bmod n)$$

где  $l$  количество запросов 2-го типа, данных до текущего запроса (сам текущий запрос не считая).

### Формат входных данных

На первой строке ввода даны два числа через пробел  $2 \leq n \leq 10^5$  – количество вершин в графе и  $1 \leq q \leq 10^5$  – количество запросов. Далее идет  $q$  строк, в каждой идёт три числа через пробел  $0 \leq a, b, c < 998244353$

### Формат выходных данных

Выведите для каждого запроса второго типа ответ на него в отдельной строке.

### Примеры

| стандартный ввод              | стандартный вывод |
|-------------------------------|-------------------|
| 6 12                          | 0                 |
| 143209915 123840720 97293110  | 2                 |
| 89822758 207184717 689046893  | 0                 |
| 67757230 270920641 26993265   | 2                 |
| 952858464 221010240 871605818 | 6                 |
| 730183361 147726243 445163345 | 0                 |
| 963432357 295317852 195433903 | 1                 |
| 120904472 106195318 615645575 |                   |
| 817920568 27584394 770578609  |                   |
| 38727673 250957656 506822697  |                   |
| 139174867 566158852 412971999 |                   |
| 205467538 606353836 855642999 |                   |
| 159292205 319166257 51234344  |                   |
| 2 1                           |                   |
| 377373366 41280240 33617925   |                   |

## Задача С. Встреча делегаций

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 1024 мегабайта

Две it-компании решили устроить встречу своих представителей для обсуждения деловых вопросов. Сотрудники в каждой компании расставили в приоритете, по которому нужно их отправлять на делегацию. Обозначим за  $a_i$  — скилл  $i$ -го приоритетного сотрудника из первой компании. Аналогично определим  $b_i$  для второй компании. На встрече планируется общение каждой пары из сотрудников обеих компаний. Известно, что чем сильнее различается скилл, тем им сложнее найти общий язык. Дискомфорт общения для пары  $(i, j)$  вычисляется как  $|a_i - b_j|$ .

Компании еще не знают, сколько людей отправит каждая, поэтому они хотят рассмотреть несколько различных вариантов. Они хотят узнавать суммарный дискомфорт для всех пар сотрудников на встрече, если первая компания хочет отправить  $x$  самых приоритетных сотрудников, а вторая компания —  $y$ . Вам, как внешнему консультанту для обеих фирм, поручили научиться это считать.

### Формат входных данных

В первой строке дано число  $n$  — количество сотрудников в каждой компании ( $1 \leq n \leq 10^5$ ).

Во второй строке дан массив целых чисел  $a_1, a_2, \dots, a_n$  — скиллы сотрудников первой компании в порядке приоритета ( $0 \leq a_i \leq 2 \cdot 10^8$ ).

В третьей строке дан массив целых чисел  $b_1, b_2, \dots, b_n$  — скиллы сотрудников второй компании в порядке приоритета ( $0 \leq b_i \leq 2 \cdot 10^8$ ).

В четвертой строке дано целое число  $q$  — количество интересующих вариантов делегаций ( $1 \leq q \leq 10^4$ ).

В следующих  $q$  строках даны пары чисел  $x_i, y_i$  — размеры делегаций первой и второй компании ( $1 \leq x_i, y_i \leq n$ ).

### Формат выходных данных

Для каждого варианта выведите суммарный дискомфорт.

### Примеры

| стандартный ввод                                             | стандартный вывод   |
|--------------------------------------------------------------|---------------------|
| 2<br>2 4<br>3 5<br>4<br>1 1<br>1 2<br>2 1<br>2 2             | 1<br>4<br>2<br>6    |
| 5<br>3 1 8 3 2<br>7 2 1 4 5<br>4<br>1 3<br>3 2<br>4 2<br>3 5 | 7<br>19<br>24<br>45 |

## Задача D. На олимпиаду

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 1024 мегабайта

Сегодня у Сергея Андреевича важное мероприятие — его ученики будут писать школьный этап ВсОШ по информатике, и именно ему предстоит решить, кого из них отправить на олимпиаду, а кого — обратно в школу на уроки.

В данный момент все  $n$  учеников Сергея Андреевича стоят в ряд, и  $i$ -му в ряду учитель присвоил положительное число  $a_i$  от 1 до  $n$  — его показатель глупости. Также у него есть  $q$  планов, каждый из которых характеризуется тремя числами  $l, r, x$  и означает, что Сергей Андреевич будет действовать по следующему алгоритму:

1. Возьмёт подотрезок школьников с  $l$ -го номера по  $r$ -й.
2. Выкидывает из рассмотрения школьников, у которых показатель глупости не менее  $x$ , как бесперспективных.
3. Некоторым образом переставляет выбранных школьников, выбирая порядок, в котором они будут сидеть на туре.

Показателем вариативности плана называется количество списков показателей глупости школьников в порядке рассадки, которые может получить Сергей Андреевич, следуя данному плану. Для статистики учитель очень хочет для каждого плана узнать его показатель вариативности. Помогите ему. Поскольку ответ может быть очень большим, он просит вас лишь посчитать его остаток по модулю 998244353.

### Формат входных данных

В первой строке даны два числа:  $n$  и  $q$  ( $1 \leq n, q \leq 25 \cdot 10^4$ ) — число учеников у Сергея Андреевича и количество планов.

Во второй строке даны  $n$  чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — показатели глупости школьников.

В следующих  $q$  строках расположены по три числа:  $l, r, x$  — параметры очередного плана ( $1 \leq l \leq r \leq n, 1 \leq x \leq n$ ).

### Формат выходных данных

Для каждого плана выведите одно число — его показатель вариативности по модулю 998244353.

### Примеры

| стандартный ввод                                                             | стандартный вывод            |
|------------------------------------------------------------------------------|------------------------------|
| 5 3<br>1 2 3 3 1<br>1 5 3<br>3 4 1<br>1 3 4                                  | 3<br>1<br>6                  |
| 8 6<br>6 2 4 1 5 1 8 6<br>5 6 3<br>1 5 7<br>1 4 6<br>4 7 8<br>4 8 2<br>5 8 6 | 1<br>120<br>6<br>3<br>1<br>2 |

## Задача Е. Линейная максимизация

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 5 секунд          |
| Ограничение по памяти:  | 1024 мегабайта    |

Имеется множество  $S$  точек на двумерной плоскости. Изначально множество  $S$  пусто. Последовательно для каждого  $i$  от 1 до  $Q$  выполните следующий запрос:

- даны целые числа  $X_i$ ,  $Y_i$ ,  $A_i$  и  $B_i$ ;
- добавьте точку  $(X_i, Y_i)$  в множество  $S$ ;
- после этого найдите значение

$$\max_{(x,y) \in S} \{A_i x + B_i y\}.$$

### Формат входных данных

В первой строке содержится целое число  $Q$  — количество запросов.

В следующих  $Q$  строках содержатся по четыре целых числа  $X_i$ ,  $Y_i$ ,  $A_i$  и  $B_i$  — координаты добавляемой точки и коэффициенты линейного выражения в запросе  $i$ .

Гарантируется, что выполняются следующие ограничения:

- $1 \leq Q \leq 2 \cdot 10^5$ ;
- $|X_i| \leq 10^9$ ;
- $|Y_i| \leq 10^9$ ;
- $|A_i| \leq 10^9$ ;
- $|B_i| \leq 10^9$ ;
- все добавляемые точки различны;
- все числа во входных данных являются целыми.

### Формат выходных данных

Выведите  $Q$  строк.

В строке  $i$  выведите значение

$$\max_{(x,y) \in S} \{A_i x + B_i y\}$$

после добавления точки  $(X_i, Y_i)$  в множество  $S$ .

## Примеры

| стандартный ввод                                                                                                       | стандартный вывод                                |
|------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| 4<br>1 0 -1 -1<br>0 1 2 0<br>-1 0 1 1<br>0 -1 1 -2                                                                     | -1<br>2<br>1<br>2                                |
| 9<br>-1 4 -8 -2<br>9 -9 -7 7<br>4 1 6 7<br>-4 -1 -4 -5<br>-9 3 -2 -6<br>-1 0 -8 5<br>-8 -5 0 0<br>8 3 0 -4<br>2 -5 2 5 | 0<br>35<br>31<br>21<br>36<br>87<br>0<br>36<br>31 |

## Замечание

Рассмотрим первый пример.

После первого запроса множество имеет вид  $S = \{(1, 0)\}$ . Для единственной точки множества значение выражения  $-x - y$  равно  $-1$ , поэтому ответ равен  $-1$ .

После второго запроса множество имеет вид  $S = \{(1, 0), (0, 1)\}$ . Максимальное значение выражения  $2x$  достигается в точке  $(1, 0)$  и равно  $2$ .

После третьего запроса множество имеет вид  $S = \{(1, 0), (0, 1), (-1, 0)\}$ . Максимальное значение выражения  $x + y$  достигается в точках  $(1, 0)$  и  $(0, 1)$  и равно  $1$ .

После четвёртого запроса множество имеет вид  $S = \{(1, 0), (0, 1), (-1, 0), (0, -1)\}$ . Максимальное значение выражения  $x - 2y$  достигается в точке  $(0, -1)$  и равно  $2$ .

## Задача F. Японский кроссворд

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2.5 секунд  
Ограничение по памяти: 256 мегабайт

Дан ряд из  $n$  квадратов, пронумерованных от 1 до  $n$ , и последовательность  $a$  длины  $n$ , состоящая из целых положительных чисел. Изначально первый квадрат покрашен в черный цвет, а остальные — в белый. На первом квадрате расположена фишка.

Вы можете повторить следующую операцию любое количество раз, возможно, нулевое:

- Если фишка находится в квадрате с номером  $i$ , выберите целое положительное целое число  $x$  и переместите фишку в квадрат  $i + a_i \cdot x$ . Обратите внимание, вы не можете сделать ход, если  $i + a_i \cdot x > n$ .
- Покрасьте квадрат  $i + a_i \cdot x$  в черный цвет.

Найдите количество различных строк, которые могут получиться после выполнения операций, по модулю 998244353.

### Формат входных данных

В первой строке вводится одно целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — длина строки и последовательности.

Во второй строке вводится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 2 \cdot 10^5$ ) — последовательность  $a$ .

### Формат выходных данных

Выведите ответ на задачу.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5<br>1 2 3 1 1   | 8                 |
| 1<br>200000      | 1                 |

### Замечание

В первом примере есть 8 возможных рядов, где в черный цвет покрашены квадраты:

- 1
- 1, 2
- 1, 2, 4
- 1, 2, 4, 5
- 1, 3
- 1, 4
- 1, 4, 5
- 1, 5

## Задача G. Substring Query

Имя входного файла: `str-qry.in`  
Имя выходного файла: `str-qry.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

У Пупы есть  $n$  строк  $S_1, S_2, \dots, S_n$ . Однажды его друг Луна пришёл в бухгалтерию и попросил его ответить на  $q$  вопросов: сколько строк среди  $S_{l_i}, S_{l_i+1}, \dots, S_{r_i}$  содержат  $P_i$  как подстроку?

Но в бухгалтерии все перепутали, и ответить на запросы за Пупу придется Вам.

### Формат входных данных

На первой строке 2 целых числа  $n, q$  ( $1 \leq n, q \leq 200\,000$ ). Каждая из следующих  $n$  строк содержит ровно одну строку  $S_i$  ( $|S_1| + |S_2| + \dots + |S_n| \leq 200\,000$ ). Последние  $q$  строк содержат запросы. Каждый запрос задаётся двумя целыми числами  $l_i, r_i$  и строкой  $P_i$ . ( $1 \leq l_i \leq r_i \leq n, |P_1| + |P_2| + \dots + |P_n| \leq 200\,000$ ) Все строки состоят из букв «a» и «b».

### Формат выходных данных

На каждый вопрос выведите одно число — ответ на вопрос.

### Примеры

| str-qry.in | str-qry.out |
|------------|-------------|
| 4 2        | 2           |
| a          | 2           |
| b          |             |
| ab         |             |
| bab        |             |
| 1 3 a      |             |
| 1 4 ab     |             |