

## Задача А. Лунки

|                         |                   |
|-------------------------|-------------------|
| 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 играть. Больше всего он любит играть в игру «Лунки». Это игра для одного игрока со следующими правилами:

Есть  $N$  лунок, расположенных в ряд, пронумерованных слева направо числами от 1 до  $N$ . У каждой лунки изначально установлена своя сила выброса (у лунки с номером  $i$  она равна  $a_i$ ). Если вбросить шарик в лунку  $i$ , то он тут же вылетит из нее и попадет в лунку  $i + a_i$ , после чего он опять вылетит и т.д.. Если же лунки с таким номером нету, то он просто вылетит за край ряда. На каждом из  $M$  ходов игрок выбирает одно из двух действий:

- Установить силу выброса лунки  $a$  равной  $b$ .
- Вбросить шарик в лунку  $a$  и посчитать количество прыжков шарика, прежде чем он вылетит за край ряда, а так же записать номер лунки, после выпрыгивания из которой шарик вылетел за край.

У Пети есть некоторые проблемы с математикой, поэтому, как Вы уже догадались, именно Вам предстоит произвести все подсчеты.

### Формат входных данных

Первая строка содержит два числа  $N$  и  $M$  ( $1 \leq N \leq 10^5$ ,  $1 \leq M \leq 10^5$ ) — количество лунок в ряду и количество ходов. Следующая строка содержит  $N$  целых положительных чисел, не превышающих  $N$  — начальные силы выброса лунок. Следующие  $M$  строк задают ходы, сделанные Петей. Каждая строка может быть двух типов:

- 0 a b
- 1 a

Тут, первый тип означает что требуется установить силу выброса лунки  $a$  равной  $b$ , а второй означает что требуется вбросить мячик в лунку с номером  $a$ . Числа  $a$  и  $b$  — целые положительные и не превышают  $N$ .

### Формат выходных данных

Для каждого хода второго типа (задающего вбрасывание шарика) в порядке следования во входном файле выведите два числа через пробел в отдельной строке — номер последней лунки перед вылетом шарика за край и количество прыжков.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 8 5              | 8 7               |
| 1 1 1 1 1 2 8 2  | 8 5               |
| 1 1              | 7 3               |
| 0 1 3            |                   |
| 1 1              |                   |
| 0 3 4            |                   |
| 1 2              |                   |

## Задача В. Треугольники

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

Дан граф. Найдите количество циклов длины 3.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 3 \cdot 10^5$ ) — количество вершин и рёбер, соответственно.

Каждая из следующих  $m$  строк содержит по два целых числа от 1 до  $n$  — вершины, которые соединяет соответствующее ребро.

Гарантируется, что в графе нет петель и кратных рёбер.

### Формат выходных данных

Выведите одно число — ответ.

### Пример

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

## Задача С. Нестабильность сети

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

Вася — системный администратор в большой компании под названием Глюкософт. В сети компании  $n$  компьютеров, и какие-то пары компьютеров напрямую соединены сетевыми кабелями, всего таких соединений ровно  $m$ , при чем никакой кабель не соединяет компьютер с самим собой, и между любой парой компьютеров не больше одного кабеля.

На каждом компьютере в Глюкософте установлена специальная программа, поддерживаемая разработчиками Глюкософта, под названием СетеБаг. Новые версии этой программы выходят почти каждый день, но система обновлений в компании совершенно непостижима. Так, обновление версии программы происходит не синхронно для всех компьютеров. Это происходит следующим образом — выбирается один компьютер, и на него устанавливается какая-то версия СетеБага. Поскольку система обновлений настроена очень странным образом, вполне возможно, что на компьютер установят не последнюю версию СетеБага, а какую-то другую, например, ту что уже установлена на нем, или даже какую-то более раннюю.

После нескольких месяцев работы, Вася обнаружил, что наиболее частая причина нестабильной работы сети в несовместимости версий СетеБага двух компьютеров в Глюкософте. Если два компьютера, напрямую соединенных кабелем, имеют установленный Сетебаг разных версий, то соединение между этими компьютерами *нестабильно*: попытка передачи данных между ними может вызвать отказ системы. Чем больше нестабильных соединений между компьютерами, тем больше вероятность отказа системы. С другой стороны, обмен данными между компьютерами с одинаковой версией СетеБага обычно не приводит ни к каким ошибкам.

Задача Васи — предотвратить и исправить ошибки в сети Глюкософта. Сделать он это может только если он находится на рабочем месте целый день. Вася хочет быть на работе, когда вероятность отказа системы достаточно высока. Он хочет распланировать свое расписание на следующий год, но оценка вероятностей отказа системы — довольно сложная задача.

Помогите Васе найти число нестабильных соединений между компьютерами после каждого обновления версий СетеБага.

### Формат входных данных

В первой строке даны два целых числа  $n, m$  — количество компьютеров и соединений между ними ( $1 \leq n, m \leq 10^5$ ).

Во второй строке даны  $n$  целых чисел  $v_1, v_2, \dots, v_n$  — версии СетеБага, изначально установленные на компьютеры Глюкософта.

В следующих  $m$  строках даны пары целых чисел  $a_i, b_i$  — номера компьютеров, соединенных  $i$ -м кабелем ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ). Гарантируется, что никакие два компьютера не соединены больше, чем одним кабелем.

В следующей строке дано целое число  $q$  — количество запланированных обновлений версий СетеБага ( $1 \leq q \leq 10^5$ ).

В следующих  $q$  строках даны пары целых чисел  $c_i, v_i$  — номер компьютера, на котором обновляется версия СетеБага, и новая версия СетеБага, которая установится на компьютер ( $1 \leq c_i \leq n, 1 \leq v_i' \leq 10^5$ ). Все обновления даны в хронологическом порядке, и никакие два обновления не происходят одновременно.

### Формат выходных данных

Выведите для каждого запроса изменения одно целое число — количество нестабильных соединений сразу после очередного обновления версии СетеБага.

## Примеры

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

## Задача D. Варенье

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

Малыш и Карлсон решили пойти на прогулку. Они знают, что прогулка будет совсем скучной, если перед ней не опустошить несколько банок варенья.

Малыш достал из кладовки  $n$  банок варенья и выставил их в ряд. В банке номер  $i$  содержится ровно  $a_i$  грамм варенья. Карлсон немного подумал и решил, что в некоторых банках недостаточно варенья, и что в банке номер  $i$  должно быть хотя бы  $b_i$  грамм варенья.

Выходить из этой ситуации Карлсон хочет в  $m$  этапов. На каждом этапе он выбирает числа  $l, r, x, y$  а затем выполняет следующие операции: в банку номер  $l$  он добавляет  $x$  грамм варенья, в банку номер  $l + 1$  добавляет  $x + y$  грамм варенья, в банку номер  $l + 2$  —  $x + 2 \cdot y$ , и так далее. В банку номер  $r$  наш герой добавит  $x + y \cdot (r - l)$  грамм варенья.

Малышу хочется определить для каждой банки  $i$  наименьший номер операции, после которой в ней станет хотя бы  $b_i$  грамм варенья. Помогите Малышу: найдите соответствующее число для каждой банки.

### Формат входных данных

В первой строке входного файла задано одно число  $n$  ( $1 \leq n \leq 10^5$ ) — количество банок. Во второй строке заданы  $n$  чисел  $a_i$  ( $0 \leq a_i \leq 2 \cdot 10^9$ ) — изначальное количество варенья в банке номер  $i$ . В третьей строке заданы  $n$  чисел  $b_i$  ( $0 \leq b_i \leq 2 \cdot 10^9$ ) — минимальное количество варенья, которое должно быть в банке номер  $i$ .

В четвертой строке задано  $m$  ( $0 \leq m \leq 10^5$ ) — число этапов добавления варенья в банки, которые выполнит Карлсон. В следующих  $m$  строках описаны сами этапы в хронологическом порядке. Каждый этап задан четырьмя числами  $l, r, x, y$  ( $1 \leq l \leq r \leq n, 0 \leq x, y \leq 10^5$ ).

### Формат выходных данных

Выведите  $n$  чисел в одной строке, разделенные пробелом. Число номер  $i$  должно быть равно 0, если в банке номер  $i$  изначально было достаточно варенья, номеру этапа, после которого в ней станет хотя бы  $b_i$  варенья, или  $-1$ , если даже после выполнения всех этапов, в этой банке будет недостаточно варенья. Этапы нумеруются с единицы.

### Пример

| стандартный ввод                                                  | стандартный вывод |
|-------------------------------------------------------------------|-------------------|
| 5<br>5 4 4 2 1<br>7 7 4 7 7<br>3<br>1 2 2 0<br>2 5 1 1<br>3 4 2 2 | 1 2 0 3 -1        |

## Задача Е. Машинное обучение

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

На курсе машинного обучения вам выдали первое домашнее задание — вам предстоит проанализировать некоторый массив из  $n$  чисел.

В частности, вы интересуетесь так называемой *равномерностью* массива. Предположим, что в массиве число  $b_1$  встречается  $k_1$  раз,  $b_2$  —  $k_2$  раз, и т.д. Тогда *равномерностью* массива называется такое минимальное целое число  $c \geq 1$ , что  $c \neq k_i$  для любого  $i$ .

В рамках вашего исследования вы хотите последовательно проделать  $q$  операций.

- Операция  $t_i = 1, l_i, r_i$  задаёт запрос исследования. Необходимо вывести равномерность массива, состоящего из элементов на позициях от  $l_i$  до  $r_i$  включительно.
- Операция  $t_i = 2, p_i, x_i$  задаёт запрос уточнения данных. Начиная с этого момента времени  $p_i$ -му элементу массива присваивается значения  $x_i$ .

### Формат входных данных

Первая строка содержит  $n$  и  $q$  ( $1 \leq n, q \leq 100\,000$ ) — размер массива и число запросов соответственно.

Во второй строке записаны ровно  $n$  чисел —  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

Каждая из оставшихся  $q$  строк задаёт очередной запрос.

Запрос первого типа задаётся тремя числами  $t_i = 1, l_i, r_i$ , где  $1 \leq l_i \leq r_i \leq n$  — границы соответствующего отрезка.

Запрос второго типа задаётся тремя числами  $t_i = 2, p_i, x_i$ , где  $1 \leq p_i \leq n$  — позиция в которой нужно заменить число, а  $1 \leq x_i \leq 10^9$  — его новое значение

### Формат выходных данных

Для каждого запроса первого типа выведите одно число — равномерность соответствующего отрезка массива.

### Пример

| стандартный ввод    | стандартный вывод |
|---------------------|-------------------|
| 10 4                | 2                 |
| 1 2 3 1 1 2 2 2 9 9 | 3                 |
| 1 1 1               | 2                 |
| 1 2 8               |                   |
| 2 7 1               |                   |
| 1 2 8               |                   |

### Замечание

Первый запрос состоит из ровно одного элемента — 1. Минимальное подходящее  $c = 2$ .

Отрезок второго запроса состоит из четырёх 2, одной 3 и двух 1. Минимальное подходящее  $c = 3$ .

Отрезок четвёртого запроса состоит из трёх 1, трёх 2 и одной 3. Минимальное подходящее  $c = 2$ .

## Задача F. МЕХ на пути

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

Дано дерево, на каждом ребре которого написано неотрицательное целое число. Вам необходимо ответить на несколько запросов вида «для данных вершин  $u$ ,  $v$  назовите наименьшее неотрицательное целое число, которое **не** встречается среди чисел, написанных на ребрах на пути от  $u$  до  $v$ ».

### Формат входных данных

Первая строка входных данных содержит два числа  $n$  и  $q$  ( $2 \leq n \leq 10^5$ ,  $1 \leq q \leq 10^5$ ), количество вершин и количество запросов.

Следующие  $n - 1$  строк содержат по три числа  $u_i, v_i, x_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $0 \leq x_i \leq 10^9$ ), которые описывают ребро дерева  $(u_i, v_i)$ , на котором написано число  $x_i$ .

Следующие  $q$  строк содержат по паре чисел  $a_j, b_j$  ( $1 \leq a_j, b_j \leq n$ ), которая обозначает запрос на пути от  $a_j$  до  $b_j$ .

### Формат выходных данных

Для каждого запроса выведите единственное число — минимальное число, которое не встречается на пути.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 7 6              | 0                 |
| 2 1 1            | 1                 |
| 3 1 2            | 2                 |
| 1 4 0            | 2                 |
| 4 5 1            | 3                 |
| 5 6 3            | 3                 |
| 5 7 4            |                   |
| 1 3              |                   |
| 4 1              |                   |
| 2 4              |                   |
| 2 5              |                   |
| 3 5              |                   |
| 3 7              |                   |

## Задача G. Запросы композиции перестановок

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

Вам дан массив  $a_1, \dots, a_n$ , состоящий из перестановок длины  $m$ .

Мы можем определить операцию  $+$  для двух перестановок  $x$  и  $y$  длины  $m$  как такую перестановку  $z = x + y$ , что  $z_i = y_{x_i}$  для всех  $1 \leq i \leq m$ . Заметьте, что порядок сложения важен.

Вам даны  $q$  запросов, каждый запрос задается двумя числами  $1 \leq l \leq r \leq n$ . Рассмотрим перестановку  $b$  длины  $m$ , такую что  $b = ((\dots((a_l + a_{l+1}) + a_{l+2}) + \dots) + a_r)$ . Тогда ответом на запрос будет являться сумма  $\sum_{i=1}^m i \cdot b_i$ . Реализуйте программу, быстро отвечающую на эти запросы.

### Формат входных данных

Первая строка содержит единственное целое число  $t$  равное количеству тестовых случаев ( $1 \leq t \leq 1000$ ). Далее следует описание  $t$  тестовых случаев, каждое в следующем формате:

Первая строка каждого описания содержит два целых числа  $n, m$  ( $1 \leq n, m \leq 10^5$  и  $1 \leq n \cdot m \leq 2 \cdot 10^5$ ). Следующие  $n$  строк содержат по  $m$  различных целых чисел  $a_{i1}, a_{i2}, \dots, a_{im}$ , разделенных пробелами ( $1 \leq a_{ij} \leq m$ ). Следующая строка содержит единственное целое число  $q$  ( $1 \leq q \leq 2 \cdot 10^5$ ). Следующие  $q$  строк содержат по два целых числа  $l, r$ , разделенных пробелами ( $1 \leq l \leq r \leq n$ ).

Гарантируется, что сумма  $n \cdot m$  и сумма  $q$  по всем тестовым случаям не превосходит  $2 \cdot 10^5$ .

### Формат выходных данных

Выведите ответы на запросы в том порядке, к которому они заданы во входных данных.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1                | 10                |
| 4 3              | 11                |
| 3 2 1            | 11                |
| 1 3 2            | 14                |
| 1 2 3            | 11                |
| 2 3 1            |                   |
| 5                |                   |
| 1 1              |                   |
| 1 4              |                   |
| 3 4              |                   |
| 3 3              |                   |
| 1 3              |                   |

## Задача Н. Сумма цифр

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

Для целых чисел  $b$  и  $n$  определим функцию  $f(b, n)$  так:

- Если  $n < b$ , то  $f(b, n) = n$
- Иначе  $f(b, n) = f(b, \text{floor}(n/b)) + (n \bmod b)$

Здесь  $\text{floor}(n/b)$  означает максимальное целое число не превышающее  $n/b$ , а  $n \bmod b$  означает остаток от деления  $n$  на  $b$ .

Менее формально  $f(n, b)$  — сумма цифр числа  $n$  в записи в системе с основанием  $b$ .

Так например:

- $f(10, 87654) = 8 + 7 + 6 + 5 + 4 = 30$
- $f(100, 87654) = 8 + 76 + 54 = 138$

Вам даны целые числа  $n$  и  $s$ . Найдите минимальное такое  $b \geq 2$ , такое что  $f(b, n) = s$ . Если такого  $b$  не существует, выведите  $-1$ .

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 10^{11}$ ). Вторая строка входных данных содержит целое число  $s$  ( $1 \leq s \leq 10^{11}$ ).

### Формат выходных данных

В единственной строке выходных данных выведите ответ на задачу.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1<br>31415926535 | -1                |
| 87654<br>30      | 10                |
| 87654<br>138     | 100               |
| 87654<br>45678   | -1                |
| 31415926535<br>1 | 31415926535       |

## Задача I. Джоске и полный граф

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

Джоске в подарок от дедушки получил огромный неориентированный взвешенный полный<sup>†</sup> граф  $G$ , который содержит  $10^{18}$  вершин. Особенность подарка в том, что вес ребра между различными вершинами  $u$  и  $v$  равен  $\gcd(u, v)$ <sup>‡</sup>. Джоске решил поэкспериментировать и сделать новый граф  $G'$ . Для этого он выбирает два целых числа  $l \leq r$ , и оставляет только такие вершины  $v$ , что  $l \leq v \leq r$ , а также оставляет только ребра между оставшимися вершинами.

Теперь Джоске интересуется, сколько различных весов ребер в графе  $G'$ . Так как граф получился огромный, он просит вашей помощи.

<sup>†</sup> Полным графом называется простой неориентированный граф, в котором каждая пара различных вершин смежна.

<sup>‡</sup>  $\gcd(x, y)$  обозначает *наибольший общий делитель (НОД)* чисел  $x$  и  $y$ .

### Формат входных данных

В первой строке содержится одно целое число  $t$  ( $1 \leq t \leq 100$ ) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа  $l, r$  ( $1 \leq l \leq r \leq 10^{18}$ ,  $l \leq 10^9$ ).

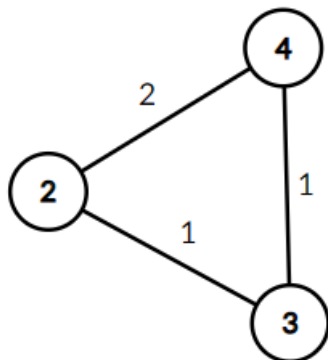
### Формат выходных данных

Для каждого набора входных данных выведите единственное число — количество различных весов среди оставшихся ребер.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 7                | 2                 |
| 2 4              | 6                 |
| 16 24            | 3                 |
| 2 6              | 5                 |
| 1 10             | 0                 |
| 3 3              | 5                 |
| 2562 2568        | 50045             |
| 125 100090       |                   |

### Замечание



Графа  $G'$  для первого набора входных данных.

На рисунке видно, что в графе 2 различных веса ребер.

В пятом наборе входных данных остается лишь одна вершина, из которой не исходит ни одного ребра, поэтому ответ — 0.

## Задача J. Лучшая пара

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

Дан массив  $a$  размера  $n$ . Пусть  $cnt_x$  — количество элементов исходного массива, равных  $x$ . Определим  $f(x, y)$  как  $(cnt_x + cnt_y) \cdot (x + y)$ .

Также даны  $m$  плохих пар  $(x_i, y_i)$ . Обратите внимание, что если  $(x, y)$  плохая пара, то  $(y, x)$  тоже плохая.

Найдите максимальное значение  $f(u, v)$  по всем парам  $(u, v)$  таким, что  $u \neq v$ , что эта пара не является плохой, а также что числа  $u$  и  $v$  встречаются в массиве  $a$ . Гарантируется, что хотя бы одна такая пара существует.

### Формат входных данных

В первой строке задано одно целое число  $t$  ( $1 \leq t \leq 10\,000$ ) — количество наборов входных данных.

Первая строка набора содержит два целых числа  $n$  и  $m$  ( $2 \leq n \leq 3 \cdot 10^5$ ,  $0 \leq m \leq 3 \cdot 10^5$ ) — длину массива и количество плохих пар.

Вторая строка набора содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — элементы массива.

$i$ -я из следующих  $m$  строк содержат два целых числа  $x_i$  и  $y_i$  ( $1 \leq x_i < y_i \leq 10^9$ ) — очередную плохую пару. Гарантируется, что ни одна пара не вводится два раза. Также гарантируется, что  $cnt_{x_i} > 0$  и  $cnt_{y_i} > 0$ .

Гарантируется, что в каждом наборе входных данных существует хотя бы одна пара чисел  $(u, v)$ ,  $u \neq v$ , не являющаяся плохой, такая, что оба этих числа встречаются в массиве  $a$ .

Гарантируется, что сумма  $n$  и сумма  $m$  не превосходят  $3 \cdot 10^5$ .

### Формат выходных данных

Для каждого набора входных данных выведите единственное целое число — ответ на задачу.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3                | 40                |
| 6 1              | 14                |
| 6 3 6 7 3 3      | 15                |
| 3 6              |                   |
| 2 0              |                   |
| 3 4              |                   |
| 7 4              |                   |
| 1 2 2 3 1 5 1    |                   |
| 1 5              |                   |
| 3 5              |                   |
| 1 3              |                   |
| 2 5              |                   |

### Замечание

В первом тестовом случае в массиве встречаются значения 3, 6, 7.

- $f(3, 6) = (cnt_3 + cnt_6) \cdot (3 + 6) = (3 + 2) \cdot (3 + 6) = 45$ . Но пара  $(3, 6)$  плохая, поэтому её не учитываем.
- $f(3, 7) = (cnt_3 + cnt_7) \cdot (3 + 7) = (3 + 1) \cdot (3 + 7) = 40$ .
- $f(6, 7) = (cnt_6 + cnt_7) \cdot (6 + 7) = (2 + 1) \cdot (6 + 7) = 39$ .

Ответ на задачу  $\max(40, 39) = 40$ .

## Задача К. Частота строки

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 1.5 секунд  
 Ограничение по памяти: 512 мегабайт

Дана строка  $s$ . Требуется ответить на  $n$  запросов.  $i$ -й запрос состоит из целого числа  $k_i$  и строки  $m_i$ , ответом является минимальная длина строки  $t$  такой, что  $t$  является подстрокой  $s$  и строка  $m_i$  входит в  $t$  как подстрока не менее  $k_i$  раз.

Подстрокой строки называется любая последовательность подряд идущих символов в этой строке.

Гарантируется, что для любых двух запросов строки  $m_i$  из этих запросов различны.

### Формат входных данных

В первой строке содержится строка  $s$  ( $1 \leq |s| \leq 10^5$ ).

Во второй строке содержится целое число  $n$  ( $1 \leq n \leq 10^5$ ).

В каждой из следующих  $n$  строк содержатся целое число  $k_i$  ( $1 \leq k_i \leq |s|$ ) и непустая строка  $m_i$  — параметры запроса с номером  $i$ .

Все строки во вводе состоят только из строчных букв латинского алфавита. Суммарная длина всех строк во вводе не превосходит  $10^5$ . Все  $m_i$  различны.

### Формат выходных данных

Для каждого запроса выведите ответ на него в отдельной строке.

Если строка  $m_i$  встречается в  $s$  менее  $k_i$  раз, выведите  $-1$ .

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| aaaaa            | 3                 |
| 5                | 4                 |
| 3 a              | 4                 |
| 3 aa             | -1                |
| 2 aaa            | 5                 |
| 3 aaaa           |                   |
| 1 aaaaa          |                   |
| abbb             | -1                |
| 7                | 2                 |
| 4 b              | -1                |
| 1 ab             | 3                 |
| 3 bb             | -1                |
| 1 abb            | 1                 |
| 2 bbb            | -1                |
| 1 a              |                   |
| 2 abbb           |                   |

## Задача L. Суровый корректор

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

По мнению Александра Павловича, текст необычайно красив, если некоторые особые слова (например, «коммунизм», «Ленин», «счастье») встречаются не слишком часто, но и не слишком редко, к тому же достаточно равномерно. Александр Павлович работает корректором. К нему поступают тексты, он имеет право их некоторым образом менять, после чего возвращает уже исправленную версию. В связи со своими воззрениями о красоте Александру Павловичу постоянно приходится проверять, сколько особых слов сейчас в той или иной части текста. Он настолько устал от рутинного подсчёта: «а сколько тут особых слов?», «а сколько тут?», что просит вас помочь ему автоматизировать этот процесс.

### Формат входных данных

Первая строка входного файла содержит текст  $s$ , в котором ищутся особые слова. Следующая строка содержит  $n$  ( $1 \leq n \leq 10^5$ ) — количество особых слов. Следующие  $n$  строк содержат особые слова. Все особые слова различны. Суммарная длина строк не превосходит  $10^5$ . В следующей строке дано  $q$  ( $1 \leq q \leq 10^5$ ) — количество интересных Александру Павловичу отрезков. Следующие  $q$  строк содержат границы отрезков  $l, r$  ( $1 \leq l \leq r \leq |s|$ ).

### Формат выходных данных

Выведите  $q$  чисел — количества вхождений особых слов в соответствующий отрезок текста.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| abacababa        | 5 0 2             |
| 2                |                   |
| a                |                   |
| aba              |                   |
| 3                |                   |
| 5 9              |                   |
| 2 2              |                   |
| 2 6              |                   |

## Задача М. Различные числа и вставки

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

Это интерактивная задача. Сразу после вывода ответа на каждый запрос необходимо сбрасывать буфер вывода. Например, в C++ используйте `fflush(stdout)` или `cout.flush()`, в Java или Kotlin — `System.out.flush()`, а в Python — `sys.stdout.flush()`.

Вам дан массив  $a$ , изначально состоящий из  $n$  целых чисел. Необходимо обработать  $q$  запросов следующих типов:

- $+ \ i \ x$  — вставить целое число  $x$  после первых  $i$  элементов массива  $a$  (при  $i = 0$  — в начало массива, при  $i = n'$  — в конец, где  $n'$  — текущая длина массива);
- $? \ l \ r$  — найти количество различных целых чисел среди  $a_l, a_{l+1}, \dots, a_r$ .

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $q$  ( $1 \leq n, q \leq 10^5$ ) — начальное количество элементов массива и количество запросов соответственно.

Во второй строке даны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — начальные элементы массива.

Каждый запрос задаётся в отдельной строке в одном из следующих форматов. Здесь  $n'$  обозначает текущую длину массива, а  $n$  — его начальную длину:

- $+ \ i \ x$  ( $0 \leq i \leq n'$ ;  $1 \leq x \leq n$ );
- $? \ l \ r$  ( $1 \leq l \leq r \leq n'$ ).

### Формат выходных данных

Для каждого запроса второго типа выведите в отдельной строке одно целое число — количество различных целых чисел среди  $a_l, a_{l+1}, \dots, a_r$ .

После вывода каждого ответа сбросьте буфер вывода, прежде чем читать следующий запрос. Иначе вы можете получить вердикт `Idleness Limit Exceeded`.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 6              | 2                 |
| 1 2 1            | 1                 |
| ? 1 3            | 3                 |
| + 0 1            |                   |
| + 0 1            |                   |
| ? 1 3            |                   |
| + 1 3            |                   |
| ? 1 6            |                   |