1. Гонки

Вася и Петя поссорились и хотят выяснить кто из них прав с помощью забега по очень длинной прямой. Организм Васи устроен так, что он может бежать с постоянной скоростью **v1**(м/с)в течение **k1** секунд, затем ему необходимо отдыхать в течение **r1** секунд, прежде чем он сможет побежать с той же скоростью снова. Аналогично Петя может бежать со скоростью **v2** в течение **k2** секунд и **r2** секунд отдыхать. Вася хочет подготовиться к забегу и просит вас определить кто из двоих мальчиков пробежит большую дистанцию в **n** забегах на различное время **t**(c) (**ti** – время i-го забега).

**Входные данные:**

В первой строке задано натуральное число **n** (1 <= n <= 106). Следующие **n** строк содержат по 7 целых чисел **v1, k1, r1, v2, k2, r2, ti** (1 <= **v1, k1, v2, k2, ti** <= 109 0 <= **r1, r2** <= 109).

**Выходные данные:**

Для каждого из t забегов выведите **V(Vasya)**, если первым к финишу придёт Вася, **P(Petya)** – если Петя, и **D(Draw)**, если забег закончится ничьёй.

**Примеры:**

|  |
| --- |
| **входные данные** |
| **3 3 1 3 1 1 2 1 1 1 2 3 1 3 1 2 1 3 1 2 2 2** |
| **выходные данные** |
| **V P D** |

1. Гипотеза

Давайте познакомимся с гипотезой Коллатца. Проделаем следующий алгоритм с натуральным числом **n**. Если оно чётное, то делим его на 2, а если нечётное, то умножаем на 3 и прибавляем 1 (получаем 3*n* + 1). Над полученным числом выполняем те же самые действия, и так далее. Гипотеза Коллатца заключается в том, что какое бы начальное число *n* мы ни взяли, рано или поздно мы получим единицу.

Доказывать мы её, конечно, не будем, однако проведём некоторые вычисления чтобы убедится в её верности для некоторых чисел.

Дано изначальное число n. Назовём его путём последовательность чисел, которые получаются в ходе преобразования числа n в единицу. Характеристиками пути назовем его длину (количество чисел в последовательности) и максимальное число (встречающееся в последовательности).

Дано **t** натуральных чисел. Для каждого из них определите характеристики его пути.

**Входные данные:**

На вход подаётся натуральное число **t** (t <= 103).  
В следующей строке содержатся **t** натуральных чисел **n** (n <= 106).

**Выходные данные:**

Для каждого входного числа **n** выведите длину его пути и максимальное число в нём.

**Примеры:**

|  |
| --- |
| **входные данные** |
| **2 4 5** |
| **выходные данные** |
| **3 4 6 16** |

**Пояснение**: путь числа пять следующий – 5 16 8 4 2 1.

1. Кусок строки

Возьмём произвольную строку s, состоящую только из строчных латинских символов. Назовём *полной* такую подстроку, которая содержит все буквы строки и при этом, является строго возрастающей (TODO). Для получения такой подстроки вы используете ваш старый компьютер, который способен только взять какую-либо подстроку фиксированной длины, а затем выполнять только две операции: заменить какой-либо символ подстроки на пустой и поменять местами любой пустой символ подстроки с любым непустым символом из оставшейся строки.

Для данной строки **s** узнайте минимальное количество операций, которые понадобятся чтобы превратить любую подстроку строки **s** в *полную* или выясните, что это невозможно.

**Входные данные:**

На вход подаётся натуральное число **n** – длина строки **s** (n <= 105).  
Затем идёт сама строка **s**.

**Выходные данные:**

Выведите одно число - минимальное количество операций, если возможно получить полную подстроку и -1 иначе.

**Примеры:**

|  |
| --- |
| **входные данные** |
| **5 abacb** |
| **выходные данные** |
| **2** |

|  |
| --- |
| **входные данные** |
| **5 bbacb** |
| **выходные данные** |
| **-1** |

**Пояснение**: в первом примере можем убрать третий символ и поставить на его место четвёртый, тогда получим полную подстроку «abc». Во втором примере ни одну из подстрок не удаётся сделать полной.

1. Галактическая налоговая

Инопланетянин X A-12 (его имя) является основателем межгалактической онлайн-школы, в которой учатся, пожалуй, жители всех планет солнечной системы. За 2024 год он заработал кругленькую сумму и теперь задумался о налогах. Как нам сообщили налоговая внимательно следит за тем, какие доходы подают бизнесмены всего Млечного Пути и считает, что бизнесмен наврал, если число, обозначающее сумму его дохода за год, удовлетворяет одному из двух условий:

* в нём встречаются подозрительные двухциферные комбинации (известные бизнесмену) из специального набора комбинаций
* в нём на некоторых (так же известных бизнесмену) позициях стоят цифры из списка подозрительных для налоговой

X A-12 не хочет подавать в налоговую доход меньше 10**n**-1 юнитов – иначе налоговая подумает что он бедный и посмеётся над ним, но и больше 10**n**– 1 тоже не хочет – так как налоговая сразу же схватит его без разбирательств. Иначе говоря, если **n** = 3, то бизнесмена устроят суммы от 100 до 999.

X A-12 хочет иметь запасной план на каждый запасной план другой запасной план, поэтому просит вас основываясь на известных ему подозрительных комбинациях и списках посчитать сколько различных сумм, не являющихся подозрительными, он может подать в налоговую.

**Входные данные:**

На вход подаётся три натуральных числа **n, m, k** (1 <= k <= n <= 105 m <= 100) – размер суммы, устраивающей X A-12, количество подозрительных комбинаций, количество позиций с подозрительными цифрами.  
В следующей строке – **m** подозрительных двухциферных комбинаций.  
И наконец **k** строк, содержащих по несколько целых чисел **i, x, a1, a2 … ax** (0 <= i < n 1 <= x < 10 0 <= a1, a2 <= 9) – позиция(справа налево), количество запрещённых на ней цифр, затем сами **x** запрещённых цифр в порядке возрастания.

**Выходные данные:**

Выведите количество различных сумм, не являющихся подозрительными.

**Примеры:**

|  |
| --- |
| **входные данные** |
| **2 2 1 12 56 0 2 1 4** |
| **выходные данные** |
| **TODO** |
|  |

**Пояснение**: среди чисел от 10 до 99 по первому условию не проходят только 12 и 56, а по второму все числа, имеющие на 0 позиции 1 или 4 (т.е. заканчивающиеся на 1 или 4).

TODO: нужен ли % 10^9?

1. Васина игра

Василий придумал из одной из прошлых задач придумал одну игру. У игрока есть массив целых чисел длины N и количество заработанных очков изначально равное нулю. Игрок делает Q ходов. На каждом ходу игрок может сделать одно из двух действий.

1. Выбрать произвольный подотрезок массива и увеличить количество своих очков на число, равное сумме максимального элемента подотрезка и индекса самого левого из максимальных элементов этого подотрезка.
2. Изменить значение элемента с индексом X на Y.

Василий, к сожалению, пока не умеет программировать и просит вас посчитать итоговое количество заработанных игроком очков.

**Входные данные:**  
На вход подаётся натуральное число **n** (n <= 10\*\*5).

Следующая строка содержит **n** целых чисел (-10\*\*9 <= ai <=10\*\*9) – элементы исходного массива.

Следующая строка содержит натуральное число **q** (q <= 10 \*\* 5) – количество ходов, сделанных игроком.

Следующие **q** строк содержат по три целых числа - ходы игрока. Первое число обозначает тип хода:

* 1 l r – ход первого типа с подотрезком [l; r] (0 <= l, r < n)
* 2 x y – ход второго типа с изменением значения элемента с индексом x на y. (0 <= x < n, -10\*\*9 <= y<=10\*\*9)

**Выходные данные:**

Выведите одно целое число - итоговое количество заработанных игроком очков.

**Примеры:**

|  |
| --- |
| **входные данные** |
| **5 1 2 3 4 3 3 1 1 4 2 3 3 1 1 4** |
| **выходные данные** |
| **12** |