

Задачи 7-9

Задача 1. count79: Количество вхождений цифры

В задаче рассмотрим две симметричные системы счисления (далее будем писать сокращённо – CCC): троичную CCC и девятеричную CCC. В троичной CCC используются цифры Z, 0, 1. К записи числа в этой системе приписывают снизу окончание $_{S3}$. $Z_{S3}=-1_{10}$, $0_{S3}=0_{10}$, $1_{S3}=1_{10}$. Если имеется запись в троичной CCC $a_n a_{n-1} \dots a_1 a_{0S3}$, то она обозначает число, равное $a_n \cdot 3^n + a_{n-1} \cdot 3^{n-1} + \dots + a_1 \cdot 3 + a_0 \cdot 1$. Например, $623_{10} = 729 - 81 - 27 + 3 - 1 = 1 \cdot 3^6 + 0 \cdot 3^5 + (-1) \cdot 3^4 + (-1) \cdot 3^3 + 0 \cdot 3^2 + 1 \cdot 3 + (-1) \cdot 1 = 10ZZ01Z_{S3}$. В девятеричной CCC используются цифры W, X, Y, Z, 0, 1, 2, 3, 4. К записи числа в этой системе приписывают снизу окончание $_{S9}$. $W_{S9}=-4_{10}$, $X_{S9}=-3_{10}$, $Y_{S9}=-2_{10}$, $Z_{S9}=-1_{10}$, $0_{S9}=0_{10}$, $1_{S9}=1_{10}$, $2_{S9}=2_{10}$, $3_{S9}=3_{10}$, $4_{S9}=4_{10}$. Если имеется запись в девятеричной CCC $a_n a_{n-1} \dots a_1 a_{0S9}$, то она обозначает число, равное $a_n \cdot 9^n + a_{n-1} \cdot 9^{n-1} + \dots + a_1 \cdot 9 + a_0 \cdot 1$. Например, $623_{10} = 729 - 81 - 27 + 2 = 1 \cdot 9^3 + (-1) \cdot 9^2 + (-3) \cdot 9 + 2 \cdot 1 = 1ZX2_{S9}$. В симметричных системах счисления перед записью числа не ставят ни плюс, ни минус, и когда число положительное, и когда оно отрицательное.

Составьте программу, которая принимает на вход в первой строке цифру J – одну из цифр девятеричной CCC {W, ..., Z, 0, 1, ..., 4}, во второй строке целое число K, записанное в троичной CCC. В этой записи используются десятичные цифры 0, 1 и заглавная латинская буква Z. Длина записи числа K в троичной CCC не более чем 100000. Программа находит, количество вхождений цифры J в запись числа K, если его перевести в девятеричную CCC. В начале записи числа K могут стоять незначащие нули, которые не следует учитывать при подсчёте количества вхождений $J=0$. Незначащим является любой ноль, стоящий левее первой ненулевой цифры, или, если $K = 0$, то все нули, кроме самого правого.

Формат входных данных

В первой строке содержится символ J – цифра девятеричной CCC (одна из {W, ..., Z, 0, 1, ..., 4}).

Во второй строке содержится непустая последовательность символов, являющаяся записью в троичной CCC целого числа K (в этой записи не более чем 100000 символов). В записи числа K используются символы из набора {Z, 0, 1}.

Формат выходных данных

В первой и единственной строке выводится неотрицательное целое число (от 0 до 50000), равное искомому количеству вхождений цифры J в запись числа K в девятеричной CCC.

Примеры

Ввод

W

000000000ZZZ

Вывод

1

Ввод

0

00000000000000000000000000000000

Вывод

1

Ввод

Y

10Z10Z10Z10Z10Z10Z10Z

Вывод

3

Задача 2. vampir79: Вампиры

В тексте будем использовать девятеричную симметричную систему счисления (сокращённо ССС), описанную в условиях задачи "Количество вхождений цифры". Рассмотрим целые положительные числа специального вида, которые назовём вампиры_{S9}. Каждое число, являющееся вампиром_{S9}, обладает всеми следующими свойствами: 1) длина его записи в девятеричной ССС является чётной (рассматривается запись без незначащих нулей в начале); 2) оно является произведением двух своих клыков -- целых положительных множителей, не равных друг другу, у которых длина записи в девятеричной ССС равна половине длины записи в девятеричной ССС вампира_{S9}; 3) если запись в девятеричной ССС одного из клыков оканчивается нулём, то у другого клыка последняя цифра в записи в девятеричной ССС не может быть нулём; 4) если выписать друг за другом слитно записи в девятеричной ССС клыков, то можно так переставить цифры в образовавшейся записи, что получится запись в девятеричной ССС вампира_{S9}. Например, $1008_{10} = 1340_{S9} = 28_{10} * 36_{10} = 31_{S9} * 40_{S9}$, значит 1008 является вампиром_{S9}. Второй пример, $81648_{10} = 134000_{S9} = 252_{10} * 324_{10} = 310_{S9} * 400_{S9}$, значит 81648 не является вампиром_{S9}, так как нарушено условие о том, что записи обоих клыков в девятеричной ССС не могут заканчиваться нулём. Третий пример, $77616_{10} = 13Y420_{S9} = 252_{10} * 308_{10} = 310_{S9} * 4Y2_{S9}$, значит 77616 является вампиром_{S9}, так как только у одного клыка запись в девятеричной ССС заканчивается нулём.

Составьте программу, которая принимает на вход в первой строке целое положительное число N, а во второй строке последовательность из N целых положительных чисел V_i, разделённых пробелами. Число N не более чем 50000. Числа V_i не более чем 2500000. Программа подсчитывает количество чисел V_i, которые являются вампирами_{S9}, и выводит результат подсчёта. Если в последовательности

более одного раза встречается одно и то же число, являющееся вампиром_{sg}, то при подсчёте учитываются все вхождения такого числа, как если бы это были вхождения разных чисел.

Формат входных данных

В первой строке содержится целое положительное число N — длина последовательности чисел, ($0 < N < 50001$).

Во второй строке содержится N целых положительных чисел V_i , разделённых пробелами ($0 < V_i < 2500001$).

Формат выходных данных

В первой и единственной строке выводится неотрицательное целое число (от 0 до 50000), равное искомому количеству вхождений во входную последовательность чисел, являющихся вампирами_{sg}.

Примеры

Ввод

1

1008

Вывод

1

Ввод

10

1 2 3 4 5 6 7 8 9 10

Вывод

0

Ввод

5

1008 77616 81648 1008 77616

Вывод

4

Задача 3. distance: Социальная дистанция

В одном университете озаботились соблюдением социальной дистанции в студенческой столовой. Зал столовой можно представить как прямоугольную сетку

размера N по вертикали на M по горизонтали, в каждой клетке которой может сесть посетитель.

В момент времени s_i , когда приходит посетитель администратор узнает момент времени f_i , когда посетитель уйдет и выбирает свободное место исходя из требований:

- место должно быть свободным
- место посетителя максимально удалено от занятых мест в зале,
- если таких мест несколько, то выбираются места с минимальной координатой y по вертикали
- если таких мест несколько, то выбираются места с минимальной координатой x по горизонтали

Расстояние между точками (y_i, x_i) и (y_j, x_j) вычисляется как $L = \text{abs}(x_i - x_j) + \text{abs}(y_i - y_j)$

Вам требуется написать программу, которая для каждого посетителя определяет место, куда ему сесть.

Формат входных данных

В первой строке задаются размеры зала N, M ($1 \leq N, M \leq 100$) и количество посетителей K ($1 \leq K \leq 5000$).

В следующих K строках информация о посетителях s_i, f_i - время прихода и ухода.

Времена до 10^9 , гарантируется что все времена различные и $s_i < f_i$. Посетители упорядочены по s_i .

Формат выходных данных

Для каждого посетителя выведите координаты y, x ($1 \leq y \leq N, 1 \leq x \leq M$). Если свободных мест в зале нет, выведите -1 -1 для этого посетителя

Примеры

Ввод

```
4 4 3
1 3
2 4
5 6
```

Вывод

```
1 1
4 4
1 1
```

Ввод

```
1 1 2
1 10
2 9
```

Вывод

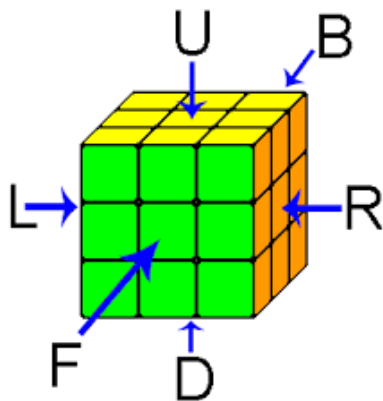
1 1
-1 -1

Задача 4. cube: Кубик

Рассмотрим кубик рубика размера $3 \times 3 \times 3$. Кубик состоит из 26 кубиков-деталей, хотя бы одна грань которых находится на поверхности кубика $3 \times 3 \times 3$. У угловых кубиков-деталей на поверхность кубика-рубика выходят 3 грани, у реберных кубиков-деталей на поверхность кубика-рубика выходят 2 грани, у центральных кубиков-деталей - 1 грань.

В центре фронтальной грани (на поверхности) кубика-рубика находится наноробот. Наноробот изначально смотрит в направлении вверх. Наноробот может исполнять команду 'F', что означает перемещение на $\frac{1}{3}$ размера кубика рубика в направлении своего движения, то есть за один шаг он попадает в центр смежного квадрата на поверхности кубика рубика. При необходимости наноробот переходит на смежную грань кубика-рубика, всегда оставаясь на его поверхности.

Кроме того наноробот может исполнять команды 'L' - повернуть на 90 градусов налево, 'R' - повернуть на 90 градусов вправо, и 'S' - остановиться. Команда 'S' гарантированно находится в конце последовательности команд.



Фронтальная грань обозначена "F".

Формат входных данных

На стандартный поток ввода подается последовательность команд без пробелов, заканчивающаяся командой 'S'.

Формат выходных данных

На стандартный поток вывода напечатайте количество кубиков-деталей (от 1 до 26), которые были посещены нанороботом.

Примеры

Ввод

FS

Вывод

2

Задача 5. poker79: Покер

В одном элитном апартамент-отеле было решено провести турнир по спортивному покеру на игровых автоматах. Для проведения турнира на все игровые автоматы загружается одинаковая последовательность конфигураций, и игроки соревнуются в том, кто выиграет больше партий. Игровые конфигурации обозначаются символами цифр '0'-'9' и строчными латинскими буквами 'a'-'z'. Специальная конфигурация, обозначаемая '#', вызывает перезагрузку автомата. Конфигурация '#' встречается ровно один раз в конце последовательности игровых конфигураций.

Игровой автомат исполняет последовательность конфигураций циклически, то есть дойдя до конфигурации '#' начинает исполнение с начала последовательности конфигураций.

Для загрузки конфигураций в автомат они кодируются следующим образом. Пусть длина последовательности игровых конфигураций равна N . Тогда, добавляя к ним конфигурацию '#', получим последовательность длины $N + 1$. Возьмем $N + 1$ начальных позиций в последовательности (от первой и до $N + 1$) и для каждой начальной позиции возьмем $N + 1$ следующих за ней конфигураций. Получим $N + 1$ последовательностей каждой длины $N + 1$.

Например, если начальная последовательность из 4 конфигураций равна:
1231

добавляя к ней '#' и беря все возможные начальные позиции, получим 5 последовательностей:

1231#
231#1
31#12
1#123
#1231

отсортируем эти последовательности лексикографически:

#1231
1#123
1231#
231#1
31#12

Мы получили матрицу из $N + 1$ строк и $N + 1$ столбцов. Теперь возьмем в этой матрице последний столбец: 13#12. Это будет закодированная последовательность конфигураций.

Ваша задача — написать программу, которая раскодирует закодированную последовательность конфигураций.

Формат входных данных

На стандартном потоке ввода задается одна строка длины не более 250 символов — закодированная последовательность конфигураций. Гарантируется, что конфигурация '#' встречается в последовательности ровно один раз.

Формат выходных данных

На стандартный поток вывода напечатайте декодированную последовательность конфигураций в виде строки. Конфигурацию '#' не печатайте.

Примеры

Ввод

13#12

Вывод

1231

Задача 6. update: Обновление

Для обновления программного обеспечения в одной р2р сети на центральный узел с номером 0 был загружен апдейт состоящий из K частей по 1 мегабайту. В сети N узлов, соединенные каждый с каждым так, что между парой узлов за одну секунду в каждом направлении может передаваться одна часть размера 1 мегабайт. Каждый узел может одновременно принимать и передавать данные. Когда какая-то часть обновления загружается полностью в узел, эта часть постоянно сохраняется в памяти узла. Каждый узел включается и начинает прием/передачу обновления в момент времени t_i от момента загрузки на центральный узел. Вам требуется определить момент времени, когда все части обновления будут загружены на все узлы сети.

Формат входных данных

В первой строке вводится N и K - число узлов в сети, включая центральный и число частей обновления ($1 \leq N \leq 10^5$. $1 \leq K \leq 10^6$). В следующих строках находятся N чисел - времена включения узлов, начиная с нулевого ($0 \leq t_i \leq 10^6$, $t_0 = 0$)

Формат выходных данных

Выведите одно число - минимальный момент времени, когда на всех узлах сети будут все части обновления

Примеры

Ввод

3 3

0

0

0

Вывод

2

Ввод

2 2

0

1

Вывод

3