

Три-одиозные числа. 7-10 класс. Заключительный этап

Рассмотрим последовательность чисел, называемых числами трибоначчи: $T_0 = T_1 = 0, T_2 = 1, T_3 = T_2 + T_1 + T_0 = 1, \dots, T_n = T_{n-1} + T_{n-2} + T_{n-3}$ для любого натурального $n \geq 3$. Числа $T_3, T_4, \dots, T_n, \dots$ можно использовать для записи любого неотрицательного целого числа в так называемой трибоначчиевой системе счисления. Эта система счисления является позиционной и использует цифры 0 и 1. В этой системе ноль записывается как 0_{tri} , один как 1_{tri} , два как 10_{tri} , три как 11_{tri} . В общем случае неотрицательное целое число $N = d_k * T_{k+3} + d_{k-1} * T_{k+2} + \dots + d_0 * T_3 = d_k d_{k-1} \dots d_0_{tri}$, где любое d_i является либо 0, либо 1, и среди любых трёх идущих подряд d_i, d_{i-1}, d_{i-2} обязательно есть хотя бы один 0, т. е. $d_i * d_{i-1} * d_{i-2} = 0$.

Неотрицательное целое число назовём три-одиозным, если в записи этого числа в трибоначчиевой системе счисления количество единиц нечётное. Примеры три-одиозных чисел: $1_{10} = 1_{tri}, 2_{10} = 10_{tri}, 733_{10} = 10101101101_{tri}$.

Составьте программу, принимающую на вход в первой строке десятичное число N – положительное натуральное число ($1 \leq N \leq 1000$) – количество чисел, а в последующих N строках – неотрицательные целые числа A_1, A_2, \dots, A_N , записанные в десятичной системе, $0 \leq A_i \leq 2^{24} - 1$. Программа подсчитывает количество чисел A_i , являющихся три-одиозными и выводит это количество в виде неотрицательного целого числа, записанного в десятичной системе без значащих нулей.

Формат ввода: В 1-ой строке содержится десятичное число N – положительное натуральное число ($1 \leq N \leq 1000$). В последующих N строках содержатся неотрицательные целые числа A_1, A_2, \dots, A_N , записанные в десятичной системе, $0 \leq A_i \leq 2^{24} - 1$. Каждое число A_i записывается в отдельной строке.

Формат вывода: Искомое количество три-одиозных чисел выводится в виде неотрицательного целого числа, записанного в десятичной системе без значащих нулей.

Ввод примера №1:

1

0

Вывод примера №1:

0

Ввод примера №2:

2

0

1

Вывод примера №2:

1

Ввод примера №3:

4

1

2

733

2

Вывод примера №3:

4

Квадродерево. 7-10 классы. Заключительный этап

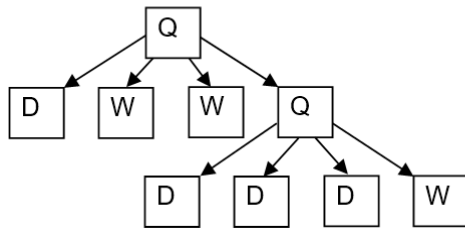
Рассмотрим способ представления плоских квадратных изображений, называемый квадродеревом:

1) Если изображение целиком одноцветное, то оно представляется квадродеревом из одной листовой вершины – пикселя. Линейная запись такого квадродерева состоит из одного символа P : W, R, O, Y, G, C, B, V или D (W – белый, R – красный, O – оранжевый, Y – жёлтый, G – зеленый, C – голубой, B – синий, V – фиолетовый или D – черный).

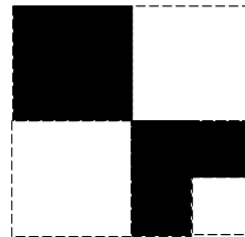
2) Если на изображении участки разного цвета, то оно делится на 4 равных квадрата (верхний левый, верхний правый, нижний левый, нижний правый) и представляется квадродеревом, состоящим из корневой вершины и четырёх поддеревьев, которые описывают части изображения – полученные при делении квадраты. Пусть $T_{ВЛД}$ – линейная запись верхнего левого поддерева, $T_{ВПД}$ – линейная запись верхнего правого поддерева, $T_{НЛД}$ – линейная запись нижнего левого поддерева, $T_{НПД}$ – линейная запись нижнего правого поддерева; тогда запись всего дерева будет такой: $QT_{ВЛД}T_{ВПД}T_{НЛД}T_{НПД}$

Пример: линейная запись квадродерева: $QDWWQDDDDW$

квадродерево в виде графа:



описываемое изображение:



В линейной записи квадродерева все символы идут подряд. Никаких пробельных, разделительных и других символов в линейной записи квадродерева не используется.

Составьте программу, на вход которой в 1-ой строке подаётся линейная запись квадродерева T (длина записи не превышает 1365 символов). Программа находит цвет, в который закрашена часть квадродерева T , имеющая максимальную площадь. Если таких цветов больше чем один, то искомым является тот, которому соответствует символ, стоящий в алфавите ближе всего к концу. Программа в первой строке выводит символ, соответствующий найденному цвету. Далее, во второй строке, полагая, что общая площадь записанного квадродеревом T изображения равна 1, программа выводит площадь части, закрашенной найденным цветом, в виде беззнаковой двоичной дроби без незначащих нулей.

Формат ввода: В 1-ой строке содержится последовательность символов $Q, W, R, O, Y, G, C, B, V$ или D , составленная по правилам линейной записи квадродерева. Длина 1-ой строки не превосходит 1365 символов.

Формат вывода: В первой строке вывода содержится единственный символ – один из W, R, O, Y, G, C, B, V или D . Он должен соответствовать цвету, в который закрашена часть квадродерева T с максимальной площадью. Если таких цветов несколько, то из их символов выбирается тот, который следует за всеми другими в алфавитном порядке. Во второй строке вывода искомая площадь части квадродерева записывается беззнаковой двоичной дробью без незначащих нулей (незначащим считается ноль, находящийся в разряде дробной части, начиная со 2-го, такой, что в части записи дроби справа от него не встречается ни одна 1). В записи вывода всегда присутствует один разряд в целой части и не менее чем один разряд в дробной. Разделителем между целой и дробной частью является точка. Например, 0 записывается так: 0.0 Пример записи, являющейся неверной из-за наличия двух незначащих нулей: 0.000100.

Ввод примера №1:

QDWWQDDDDW

Вывод примера №1:

W

0.1001

Ввод примера №2:

R

Вывод примера №2:

R

1.0

Ввод примера №3:

QWRYO

Вывод примера №3:

Y

0.01

Гномы сундуки. 7-10 классы. Заключительный этап

В сокровищнице короля гномов Кхазад-Дума Дурина хранится множество сундуков. Сундуки занумерованы, начиная с единицы. В сундуках лежат гномьи деньги. О каждом сундуке известна денежная сумма, хранящаяся в нём, – неотрицательное целое число. Гномы пользуются своей собственной системой счисления для записи номеров сундуков и денежных сумм. Число 0 по-гномьи записывается двумя символами $()$ – открывающей и закрывающей круглыми скобками. Цифра 1 по-гномьи записывается закрывающей угловой скобкой (или знаком «больше») $>$. Цифра 2 по-гномьи записывается двумя символами $>!$ – закрывающей угловой скобкой $>$ и восклицательным знаком. Цифра 3 по-гномьи записывается тремя символами $>!!$ – закрывающей угловой скобкой $>$ и двумя восклицательными знаками. Цифра 4 по-гномьи записывается четырьмя символами $>!!!$ – закрывающей угловой скобкой $>$ и тремя восклицательными знаками. Цифра 5 по-гномьи записывается двумя символами $>?$ – закрывающей угловой скобкой $>$ и вопросительным знаком. Цифра 6 по-гномьи записывается открывающей угловой скобкой (или знаком «меньше») $<$. Цифра 7 по-гномьи записывается двумя символами $<!$ – открывающей угловой скобкой $<$ и восклицательным знаком. Цифра 8 по-гномьи записывается тремя символами $<!!$ – открывающей угловой скобкой $<$ и двумя восклицательными знаками. Цифра 9 по-гномьи записывается четырьмя символами $<!!!$ – открывающей угловой скобкой $<$ и тремя восклицательными знаками. Цифра десять (A в позиционных системах счисления с основанием больше десяти) по-гномьи записывается двумя символами $<?$ – открывающей угловой скобкой $<$ и вопросительным знаком. Другие цифры (в том числе цифру 0) гномы не используют.

Ещё одной особенностью гномьей системы счисления является то, что их цифры в записи числа следуют в обратном порядке, если сравнивать с обычной позиционной системой счисления. Пусть имеется гномья запись числа $d_0d_1\dots d_{n-1}d_n$, чтобы узнать, какое ненулевое число записано, нужно вычислить: $val(d_0) * 10^0 + val(d_1) * 10^1 + \dots + val(d_{n-1}) * 10^{n-1} + val(d_n) * 10^n$. Здесь val обозначает взятие значения цифры и для удобства определено в таблице:

цифра d_i	$>$	$>!$	$>!!$	$>!!!$	$>?$	$<$	$<!$	$<!!$	$<!!!$	$<?$
значение $val(d_i)$	1	2	3	4	5	6	7	8	9	10

Получается, что от десятичной системы счисления гномья система отличается порядком записи цифр числа и тем, что в ней не используется нулевая цифра, но используется цифра десять.

Примеры:

$$0_{10} = ()$$

$$10_{10} = <?$$

$$2024_{10} = 4 * 10^0 + 2 * 10^1 + 10 * 10^2 + 1 * 10^3 = >!!!>!<?>$$

Король Дуриин хочет найти в своём подвале два сундука, таких, чтобы разность денежных сумм, лежащих в них, была максимальной. Если в подвале есть несколько искомым пар сундуков, то король Дуриин выбирает из них такую пару, что сумма номеров сундуков минимальна.

Составьте программу, принимающую на вход в первой строке десятичное число N – положительное натуральное число ($2 \leq N \leq 500$), записанное без знака в десятичной системе счисления – количество сундуков, а в последующих N строках – записи в гномьей системе денежных сумм, хранящихся в сундуках с номерами от 1 до N . Известно, что в каждой записи денежной суммы не более чем 100 символов (угловых или круглых скобок, восклицательных или вопросительных знаков). Программа находит номера K и L такие, что $K < L$, абсолютное значение (модуль) разницы денежных сумм в K -ом и L -ом сундуках наибольшее и сумма $K + L$ наименьшая. Программа выводит в первой строке гномью запись найденного числа K , а во второй строке – гномью запись найденного числа L . Каждое число записывается в гномьей системе.

Формат ввода: В первой строке содержится десятичное число без знака N – количество сундуков ($2 \leq N \leq 500$). В следующих N строках содержатся записи в гномьей системе денежных сумм, хранящихся в сундуках, – последовательности, в которых могут встретиться только угловые или круглые скобки $<$, $>$, $()$ и восклицательные или вопросительные знаки. Эти последовательности записаны по правилам составления записей чисел в гномьей системе. Длины строк находятся в диапазоне от 1 до 100 включительно.

Формат вывода: В первой строке выводится гномья запись найденного числа K . Во второй строке выводится гномья запись найденного числа L . Числа K и L таковы, что $K < L$, абсолютное значение (модуль) разницы денежных сумм в K -ом и L -ом сундуках наибольшее и сумма $K + L$ наименьшая.

Ввод примера №1:

Ввод примера №2:

Ввод примера №3:

2

()

>!!!>!<?>

Вывод примера №1:

>

>!

3

<?>

<?>

>!!!>!<?>

Вывод примера №2:

>

>!!

4

>!!!>!<?>

<?>

<?>

()

Вывод примера №3:

>

>!!!

4

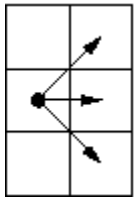
Problem 4: 4-710: Грибы отсюда

Prob Id: 4
Full score: 100
Input file name: input.txt or standard input
Time limit: 1 s
Memory RSS limit: 256M
Stack limit: 256M

Problem 4: Грибы отсюда

Гарри, Рон и Гермиона занимаются сбором грибов в тайной комнате. Тайная комната представляет собой прямоугольник из клеток размером R строк на C столбцов, в каждой клетке которого находится некоторое количество грибов.

Гарри, Рон и Гермиона начинают движение с некоторых (возможно совпадающих) позиций в первом столбце. Каждый персонаж на каждом ходу переходит либо на клетку справа от текущей клетки, либо на клетку по диагонали справа сверху от текущей клетки, либо на клетку по диагонали справа снизу от текущей клетки. Несколько персонажей могут находиться в одной клетке. Закончить движение они должны в каких-то позициях (возможно совпадающих) в последнем столбце.



Когда персонаж проходит через клетку, он собирает все грибы, которые в ней находятся, но когда несколько персонажей находятся в одной клетке, они как-то делят все грибы между собой, то есть (естественно), грибы собрать можно только один раз.

Напишите программу, подсчитывающую максимальное количество грибов, которые могут собрать друзья, пройдя через комнату.

Input format

В первой строке входных данных находятся два целых числа R ($1 \leq R \leq 20$) и C ($2 \leq C \leq 200$) — число строк и столбцов в прямоугольнике, описывающем комнату. Во второй строке входных данных находятся три целых числа r_1, r_2, r_3 — номера строк в самом левом столбце, задающие клетки, в которых в начальный момент времени находятся ребята ($0 \leq r_1, r_2, r_3 < R$).

Далее записываются $R * C$ целых чисел $f_{i,j}$ ($0 \leq f_{i,j} \leq 100$), задающих количество грибов в каждой клетке. Прямоугольник комнаты описывается по строкам, то есть сначала идут C чисел, описывающих строку 0 прямоугольника, а последними идут C чисел, описывающих строку $R-1$ прямоугольника. Числа могут разделяться произвольным числом пробельных символов, то есть переводов строк, пробелов и табуляций.

Output format

Выведите целое число — максимальное количество грибов, которых могут собрать ребята, пройдя через комнату.

Examples

Input

```
5 10
0 2 4
0 0 0 0 0 0 0 0 10 0
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3
0 0 0 10 0 0 0 0 0 0
```

Output

74

ejudge 3.12.0+ (GIT 7bbe05192) (2024-02-05 23:59:28).

Copyright © 2000-2024 Alexander Chernov.

5 Problem 5: 5-710: Транспарант

Prob Id:	5
Full score:	100
Input file name:	input.txt or standard input
Time limit:	1 s
Memory RSS limit:	256M
Stack limit:	256M

Problem 5: Транспарант

На четвёртый день плавания на теплоходе Скрябин Остапу было поручено нарисовать агитационный транспарант. В этот день теплоход остановился в Чебоксарах, где, к удаче Остапа, он увидел уже готовый транспарант, висящий на пристани. Остап решил под покровом темноты вырезать из висящего транспаранта один сплошной кусок, который незаметно занести на теплоход, и потом спокойно разрезать этот кусок по отдельным буквам, чтобы сложить полученные буквы в нужный ему агитационный транспарант. Остап хочет вырезать кусок минимальной длины из готового транспаранта, но на котором будет все буквы, необходимые для его транспаранта.

Input format

В первой строке ввода записана строка текста — уже готовый транспарант. Во второй строке ввода записана строка текста — требуемый Остапу транспарант. Длины строк не превышают 50000 символов.

В строке допускаются любые символы с ASCII-кодами 33 - 126.

Output format

Выведите строку, являющуюся сегментом (то есть фрагментом, идущим подряд) первой строки ввода, которая содержит все требуемые буквы для второй строки ввода, и имеет минимальную длину. Если таких строк несколько, выведите любую. Если такой строки не существует, выведите пустую строку.

Examples

Input

```
сасасбсас  
аб
```

Output

```
acb
```

Input

```
сасасбсас  
bab
```

Output

Notes

ejudge 3.12.0+ (GIT 7bbe05192) (2024-02-05 23:59:28).

Copyright © 2000-2024 Alexander Chernov.

Результат работы на втором тесте — пустая строка.