

Задача А. Оптимизация

Имя входного файла: `bitty.in`
Имя выходного файла: `bitty.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В компании KTP SOLUTIONS программисты часто сталкиваются с различными оптимизационными задачами. Вася очень рад задачам такого рода и прилагает массу усилий, чтобы его решения были отточены и отполированы до мелочей. В основном ему приходится работать с оптимизацией данных, поэтому он провел много времени, сжимая данные с помощью различных ухищрений над битами.

Решая одну из таких оптимизационных задач, Вася свёл её к следующей:

1. дано целое положительное число n ,
2. необходимо посчитать количество чисел k ($1 \leq k \leq n$) таких, что

$$k \oplus 2k \oplus 3k = 0.$$

Формат входного файла

Во входном файле содержится единственное число n ($1 \leq n \leq 10^{18}$).

Формат выходного файла

В выходной файл выведите целое число — количество чисел k , удовлетворяющих описанному выше условию.

Примеры

<code>bitty.in</code>	<code>bitty.out</code>
3	2
9	6

Примечание

Операция \oplus — это операция “исключающее или”, которая применяется побитово к разрядам чисел. Таблица истинности определяет для всевозможных пар битов, какое значение будет принимать операция исключающего или для них:

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

Рассмотрим пример: $5 \oplus 3$. В битовом представлении число 5 представляется как 101, число 3 — 11. Поразрядное применение исключающего или (начиная с младших разрядов) дает:

$$\begin{array}{r} 101 \\ 011 \\ \hline 110 \end{array}$$

Таким образом, $5 \oplus 3 = 6$.

Пояснение ко второму примеру

Во входном файле содержится n равное 9. Для этого числа подходят следующие значения k , которые дают верное тождество: 1, 2, 4, 5, 8, 9.

Задача В. DnB

Имя входного файла: `dnb.in`
Имя выходного файла: `dnb.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Многим из нас нравится слушать музыку работая за компьютером. Программист Вася во время работы постоянно включает и слушает электронную музыку жанра Drum 'n' Bass. Она ему позволяет сконцентрироваться на задачах и максимально быстро решить их.

После решения нескольких сотен задач на благо компании KTP SOLUTIONS Вася заинтересовался, как можно определять такого рода музыку. Первое, что пришло в голову Васе — мелодия DnB состоит из повторяющихся кусочков звуков. Однако, он понял, что это не совсем верно, и уточнил определение.

Вася записывает мелодию в виде массива a из n чисел, в котором элемент a_i соответствует амплитуде колебания звука в i -ый момент времени.

Вася считает, что мелодия относится к жанру DnB, если её можно поделить на k отрезков одинаковой длины таких, что каждый отрезок является копией предыдущего с не более чем двумя измененными величинами колебания звука в некоторых моментах времени (при этом, отрезки звуков не должны быть полностью различны!).

Формат входного файла

В первой строке входного файла содержится целое число n — количество временных единиц, в которые известны амплитуды колебания ($1 \leq n \leq 100'000$). В следующей строке находится n целых чисел — описание мелодии в виде набора амплитуд колебания, i -ое из этих чисел a_i соответствует амплитуде колебания в i -ый момент времени ($-10'000 \leq a_i \leq 10'000$).

Формат выходного файла

В выходной файл выведите ‘NO’, если данная мелодия не является мелодией жанра DnB, иначе выведите ‘YES’ и в следующей строке количество k отрезков, на которые мелодию надо разделить ($2 \leq k < n$). Если существует несколько значений k , которые относят мелодию к жанру DnB, выведите любое из них.

Примеры

<code>dnb.in</code>	<code>dnb.out</code>
18 1 2 1 2 1 3 2 3 2 3 2 3 3 3 3 2 1 2	YES 9
10 1 2 3 4 5 6 7 8 9 10	NO

Задача C. HARD

Имя входного файла: `hard.in`
Имя выходного файла: `hard.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Стажировка в компании KTP SOLUTIONS начинается с простой задачки.

Числовой матрицей $N \times M$ называют таблицу из чисел, состоящую из N строк и M столбцов. Если саму матрицу обозначить буквой A , то элемент матрицы, находящийся в i -той строке на j -том месте, обозначают A_{ij} .

Кандидату даётся матрица A размера $N \times M$ и его просят найти две матрицы B и C размера $N \times M$ такие, что, если сложить их соответствующие элементы, то получится исходная матрица (то есть $A_{ij} = B_{ij} + C_{ij}$). Всё было бы легко, но есть одно ограничение — в искомых матрицах B и C все числа должны быть различны (хотя числа из разных матриц могут и совпадать).

Сможете ли вы решить такую задачу?

Формат входного файла

В первой строке входного файла находятся два целых числа N и M ($1 \leq N, M \leq 1'000$) — размеры матрицы A . Далее в N строках находится по M целых чисел — элементы матрицы. Числа не превосходят $1'000'000$ по модулю.

Формат выходного файла

В выходной файл выведите $2 \cdot N$ строк по M чисел в каждой — искомые матрицы B и C . Элементы каждой матрицы должны быть различны и не превышать $1'000'000'000$ по модулю. Сначала выведите первую матрицу, потом вторую.

Примеры

hard.in	hard.out
2 2 1 2 3 4	0 2 -1 1 1 0 4 3
1 3 1 3 5	11 23 35 -10 -20 -30
3 3 2 4 8 16 32 64 128 256 512	1 2 3 4 5 6 7 8 9 1 2 5 12 27 58 121 248 503

Примечание

Пояснение к первому примеру:

$$\begin{pmatrix} 0 & 2 \\ -1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 4 & 3 \end{pmatrix} = \begin{pmatrix} 0+1 & 2+0 \\ -1+4 & 1+3 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Решения, работающие только со значениями элементов матриц и параметров N, M не превышающими 100, могут набрать не более 30 баллов.

Задача D. Распознавание смайликов

Имя входного файла: smile.in
Имя выходного файла: smile.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Сотрудники компании KTP SOLUTIONS работают над созданием адаптивного искусственного интеллекта, способного распознавать эмоциональное состояние собеседника во время онлайн-переписки. Одна из ключевых функций этого алгоритма — распознавание смайликов в тексте. Смайлики, будучи записанными всего двумя символами, таят в себе огромную информацию об эмоциональном состоянии личности.

Функция уже написана, но, как известно, любая часть программы, любая функция, всегда должна быть протестирована. Вас попросили помочь написать альтернативную программу, на основе результатов которой и будет проводиться тестирование. Более точно, вас попросили написать программу, которая по заданному словарю смайликов будет заменять смайлики в тексте на соответствующие слова, обозначающие эмоции.

Формат входного файла

В первой строке входного файла находится два целых числа N и M ($1 \leq N \leq 150, 1 \leq M \leq 500$).

В последующих N строках задаётся словарь смайликов. Каждая строка состоит из двух слов разделённых пробелом. Первое слово — смайлик, состоит из двух символов. Первый символ всегда “:” или “=”, второй может быть любым символом с ASCII-кодом от 33 до 126. Второе слово длиной не более 15 символов и состоит из строчных букв английского алфавита. Все смайлики различны, но разные смайлики могут означать одно и то же слово.

В последующих M строках записан некий текст для расшифровки. Каждая строка состоит из слов, разделённых пробелами. Каждое слово состоит из символов с кодами от 33 до 126. Длина каждого слова не более 15 символов. Длина каждой строки не более 500 символов. Смайликом может быть только слово из двух символов, присутствующее в словаре.

Формат выходного файла

В выходной файл выведите M строк исходного текста, где каждый найденный смайлик заменён на соответствующее ему слово из словаря. При замене добавьте в начало слова символ “\”.

Примеры

smile.in
3 2 :) smile =(sad :D laugh I'm proud :D of making this problem :) But it is too late now =(I want to sleep :(
smile.out
I'm proud \laugh of making this problem \smile But it is too late now \sad I want to sleep :(
smile.in
2 1 =/ smile_1 =0 smile_2 This string contains =/ and =0 but don't contain \smile_3!!!
smile.out
This string contains \smile_1 and \smile_2 but don't contain \smile_3!!!

smile.in

```
3 3
:D :D
:$ money_in_eyes
== ::
Smiles can be strange. Take a look on == and :D
If we test this AI function, we will be REACH!!! :$ :$
:D :D :D :D :D :D
```

smile.out

```
Smiles can be strange. Take a look on \:: and \:D
If we test this AI function, we will be REACH!!! \money_in_eyes \money_in_eyes
\:D \:D \:D \:D \:D \:D
```

Задача Е. Странная игра

Имя входного файла: `strange.in`
Имя выходного файла: `strange.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вова и Саша, сотрудники компании KTP SOLUTIONS, во время обеденного перерыва любят играть в одну странную игру. Игра заключается в следующем.

Дано некоторое натуральное число X , а также два целых числа: A и B . Первым ходом Вова вычитает единицу из числа X . Вторым ходом Саша вычитает число $1 + B$ из числа X . На третьем, пятом, седьмом и т.д. ходу Вова вычитает на A больше, чем вычел Саша в предыдущий ход. На четвертом, шестом, восьмом и т.д. ходу Саша вычитает на B больше, чем вычел Вова в предыдущий ход. Игра заканчивается, когда после очередного хода число X станет не положительным. Тот, кто сделал этот ход, объявляется победителем. Если игра никогда не сможет закончиться, то объявляется ничья.

Обеденный перерыв обычно короткий, так как работы очень много, поэтому часто начатая игра остаётся незавершённой.

Уборщица, придя на уборку помещения, заметила на столе начатую партию игры. Теперь её мучает вопрос, кто же победит в этой игре. Помогите ей это выяснить!

Формат входного файла

В первой строке входного файла заданы три целых числа — X , A , B ($2 \leq X \leq 1'000'000'000'000'000$, $-15 \leq A, B \leq 15$).

Формат выходного файла

В случае, когда игра не может быть завершена, выведите только одно слово ‘DRAW’. Если же игра закончится, то в первую строку выходного файла выведите слово ‘VOVA’, если Вова закончит игру своим ходом, иначе, выведите слово ‘SASHA’. На второй строке выведите номер хода, на котором игра закончится.

Примеры

<code>strange.in</code>	<code>strange.out</code>
3 1 1	SASHA 2
13 2 3	SASHA 4
177 2 -5	DRAW

Примечание

Во втором тесте Вова вычитает 1, получая число 12. После этого Саша вычитает $1 + 3 = 4$, получая 8. Затем Вова вычитает $4 + 2 = 6$, получая 2. После этого Саша вычитает $6 + 3 = 9$, получает -7 и становится победителем.

Решения, работающие только для $X \leq 1'000'000'000$, будут оцениваться не более, чем в 60 баллов.

Решения, работающие только для положительных A и B , будут оцениваться не более, чем в 40 баллов.

Решения, работающие только для случаев, когда игра не может быть завершена, будут оцениваться не более, чем в 20 баллов.