

Задача А. Конец света

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

Много-много лет назад древние Майя предсказали, что вчера должен был произойти конец света. Свято веря в это предсказание, простой работник телеканала «Последний» по имени Петя решил, что в последний день своей жизни терять нечего, и выкрикнул прямо в лицо директору телеканала Стрэнгу Льву Константиновичу: «MTV!», оставив того в полном недоумении.

К счастью или к несчастью, конец света так и не наступил, а Петя теперь оказался в крайне неловком положении. Чтобы срочно загладить свою вину, Пете ничего не остаётся, кроме как кемпного приврать. Он решил сказать начальнику, что имел в виду вовсе не популярный телеканал, а что-нибудь другое. Например: «Мы тобой восхищаемся!» или «Мощное телевидение возглавляешь!». Более того, Пете не просто нужно придумать расшифровку аббревиатуры «MTV» — ему также нужно, чтобы она состояла из трёх подряд идущих слов из списка любимых слов Льва Константиновича.

Строго говоря, у Пети есть список из n любимых слов Льва Константиновича. Ему интересно узнать, сколько существует различных способов придумать расшифровку аббревиатуры «MTV», если слова в расшифровке должны принадлежать списку любимых слов и должны идти в нём подряд ровно в порядке: слово на букву «М», слово на букву «Т», слово на букву «V». Более того, в списке любимых слов Льва Константиновича некоторые слова могут повторяться. Поэтому два способа расшифровки, состоящие из одинаковых слов, но стоящих на разных позициях в списке, считаются различными. Список слов может оказаться довольно длинным, так что Петя просит вас помочь ему решить эту задачу.

Формат входных данных

В первой строке дано единственное число n ($1 \leq n \leq 10^5$) - количество чисел в списке любимых слов Льва Константиновича.

В следующих n строках перечислены слова списка, по одному на строке.

Гарантируется, что сумма длин слов во входных данных не превосходит 10^5 .

Формат выходных данных

Выведите единственное число - количество возможных расшифровок аббревиатуры «MTV».

Примеры

стандартный ввод	стандартный вывод
6 Mi Toboi Voshishaemsysa Moshnoe Televidenie Vozglavlyaesh	2
10 Agniev Mi Toboi Voshishaemsysa moshnoe televidenie vozglavlyaesh Mi toboi Voshishaemsysa	1
6 Mi Toboi Voshishaemsysa Mi Toboi Voshishaemsysa	2

Задача В. Очень интересная новость

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

В некоторой социальной сети зарегистрировано n пользователей. Однажды Миша услышал очень интересную новость и решил выложить её на своей странице в этой социальной сети. Известно, что i -й пользователь просматривает страницы людей, на которых он подписан, в моменты времени t такие, что $t-i$ делится на n . Если человек впервые видит на чей-либо странице очень интересную новость, он незамедлительно выкладывает её на своей странице. Для каждого пользователя найдите момент времени, когда он выложит очень интересную новость на своей странице.

Формат входных данных

В первой строке даны два целых числа n и m ($1 \leq n, m \leq 200000$) — количество пользователей и количество подписок. Миша имеет номер 1 и выкладывает новость в момент времени 1.

Каждая из следующих n строк содержит два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), означающие, что пользователь u_i подписан на пользователя v_i .

Формат выходных данных

Выведите n целых чисел $d_1, \dots, d_n$, где d_i — время, когда i -й пользователь выложит очень интересную новость, или -1 если этого не произойдет никогда.

Примеры

стандартный ввод	стандартный вывод
4 4 2 1 3 2 4 3 1 4	1 2 3 4
4 4 1 2 2 3 3 4 4 1	1 10 7 4
3 2 2 1 2 3	1 2 -1

Задача С. Костюмы для актеров

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

Актерская труппа одно очень известного театра готовится к премьере. Они уже давно придумали сценарий, распределили роли, каждый актер уже даже выучил свой текст! Осталось разобраться только с костюмами.

Выбор подходящих костюмов - вовсе не простая задача, как может показаться. Каждый актер порой вынужден переодеваться прямо во время спектакля! Конечно, это требует времени, так что иногда успеть вовремя выйти на сцену в новом костюме гораздо труднее, чем правильно сыграть свою роль.

К счастью, в премьерной постановке каждый костюм будет состоять из не более чем 30 зашнурованных от 0 до 29 частей, каждая из которых покрашена либо в белый, либо в черный цвет. Актеры заметили, что чем сильнее отличается новый костюм от уже надетого, тем дольше придется переодеваться. Чтобы как-то оценить, какие костюмы труднее всего надевать, а какие легче всего, они решили сопоставить каждому костюму свое число по следующему принципу: пусть части костюма с номерами $y_1, y_2, \dots, y_k$ имеют черный цвет, тогда данному костюму соответствует число $2^{y_1} + 2^{y_2} + \dots + 2^{y_k}$.

Введя таким образом соответствие между числами и костюмами, актеры заметили, что время переодевания одного костюма в другой равно битовому исключающему ИЛИ (т.е. хог-у) соответствующих им чисел. Например, если один костюм соответствует числу 6, а другой числу 5, то поменять один из них на другой актер может за $6 \oplus 5 = 3$ секунды (так как $6 = 2^2 + 2^1$, а $5 = 2^2 + 2^0$, то переодеть потребуется части с номерами 0 и 1, что займет $2^0 + 2^1 = 3$ секунды); в случае же костюмов с номерами 15 и 21 на переодевание потребуется $2^4 + 2^3 + 2^1 = 24$ секунд (ибо $15 = 2^3 + 2^2 + 2^1 + 2^0$, $21 = 2^4 + 2^2 + 2^0$).

Теперь им интересно узнать для каждого имеющегося костюма: если актер наденет этот костюм, то какое наименьшее и наибольшее время на переодевание он может потратить? Так как вся труппа очень занята репетициями, они попросили вас помочь им решить эту задачу.

Формат входных данных

В первой строке находится единственное целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество костюмов, имеющихся в театре.

Во второй строке через пробел перечислены n чисел, соответствующие имеющимся костюмам: $x_1, x_2, \dots, x_n$ ($0 \leq x_i < 2^{30}$).

Формат выходных данных

Выведите n строк, i -я из которых должна содержать два числа, записанных через пробел: минимальное и максимальное время, необходимое для переодевания i -го костюма.

Примеры

стандартный ввод	стандартный вывод
2	18 18
58 40	18 18
3	0 15
8 7 8	15 15
	0 15

Задача D. Сосиска в массиве

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Петя купил в мясном отделе два массива a и b длины n . Сосиской называется такая пара индексов l, r , $l \leq r$, что существует такие позиции u , ($l \leq u \leq r$) и d , ($l \leq d \leq r$), что подмассив a с l по u не убывает, подмассив a с u по r не возрастает, подмассив b с l по d не возрастает и подмассив b с d по r не убывает. Число $r - l + 1$ называется размером сосиски.

Подмассивом массива m с l по r называется массив m' длины $r - l + 1$ такой, что $\forall i, 1 \leq i \leq r - l + 1 : m'[i] = m[i + l - 1]$.

Пете стало интересно, каков максимальный размер сосиски в его массивах.

Формат входных данных

В первой строке входного файла содержится одно целое число n , $1 \leq n \leq 10^5$ - длина Петиних массивов. Во второй строке записаны n целых чисел a_i , $|a_i| \leq 10^9$ - массив a . В третьей строке записаны n целых чисел b_i , $|b_i| \leq 10^9$ - массив b .

Формат выходных данных

Выведите размер наибольшей сосиски в массивах Пети.

Примеры

стандартный ввод	стандартный вывод
4 2 5 1 3 1 4 3 2	2
16 1 3 4 5 5 4 3 1 1 3 4 5 5 4 3 1 -1 -3 -4 -5 -5 -4 -3 -1 -1 -3 -4 -5 -5 -4 -3 -1	9

Задача Е. Петя, треугольники и контекст

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

Петя участвует в работе жюри одной Очень Известной Олимпиады по Информатике (ОИОИ). В круг обязанностей Пети входит подготовка задачи со следующим условием:

«Вам даны n различных точек на плоскости ($0 \leq n \leq 1000$). Выведите количество равнобедренных треугольников с вершинами в этих точках?».

Подготовка задачи включает в себя в том числе и подготовку тестов, и Петя обратился к Вам за помощью в этом деле. Сейчас Петя хочет сделать тест, ответ к которому будет ровно m , то есть будет существовать ровно m невырожденных равнобедренных треугольников с вершинами в этих точках. Ваша задача заключается в том, чтобы придумать такой тест.

Формат входных данных

В первой строке дано единственное число m ($0 \leq m \leq 10^5$) — требуемое количество равнобедренных треугольников.

Формат выходных данных

В первой строке выведите единственное целое неотрицательное число n , не превосходящее 1000 — количество точек в тесте.

В последующих n строках выведите координаты точек. На $i + 1$ -й строке выведите через пробел 2 целых числа x_i и y_i — координаты i -й точки. Координаты не должны превосходить 10^4 по модулю, точки должны быть попарно различны.

Гарантируется, что ответ всегда существует.

Примеры

стандартный ввод	стандартный вывод
1	3 0 0 0 1 1 0
5	5 0 0 3 -4 3 4 5 0 6 3

Задача F. Сортировка слиянием

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

Недавно Илья на лекции по программированию узнал про сортировку слиянием. Идея её проста - разбиваем массив на два примерно одинакового размера, каждый рекурсивно сортируем и затем сливаем при помощи двух указателей. Но заинтересовала его не реализация, а странная фраза преподавателя: "Сортировка работает, используя не более $O(n \cdot \log(n))$ сравнений". Илья не совсем понял доказательство этого факта и поэтому решил проверить его на практике.

Он написал следующий код:

```
cnt := 0 //Количество сравнений во время сортировки
function letter (a, b):
    cnt := cnt + 1
    return a < b

array a[MAXN] //Массив, который надо отсортировать
array buffer[MAXN] //Вспомогательный массив

procedure copy_to_buffer(l, r, ptr_buffer): //Копирует полуинтервал [l, r) из a в buffer
    while (l < r)
        buffer[ptr_buffer] := a[l]
        ptr_buffer := ptr_buffer + 1
        l := l + 1

procedure merge_sort(l, r): //В массиве a сортирует полуинтервал [l, r)
    n := r - l

    if (n <= 1) //Если массив имеет размер 1 или 0, то он уже отсортирован
        return

    m := l + n / 2 //Деление целочисленное, с округлением вниз
    merge_sort(l, m) //Сортируем левый и правый массивы
    merge_sort(m, r)

    ptr_left := l
    ptr_right := m
    ptr_buffer := l

    while (ptr_left < m and ptr_right < r) //Сливаем массивы при помощи двух указателей
        if (letter(a[ptr_left], a[ptr_right]))
            buffer[ptr_buffer] := a[ptr_left]
            ptr_left := ptr_left + 1
            ptr_buffer := ptr_buffer + 1
        else
            buffer[ptr_buffer] := a[ptr_right]
            ptr_right := ptr_right + 1
            ptr_buffer := ptr_buffer + 1

    copy_to_buffer(ptr_left, m, ptr_buffer) //Сливаем то, что осталось в левом
    copy_to_buffer(ptr_right, r, ptr_buffer) //Сливаем то, что осталось в правом
    //Хотя бы в одном массиве не осталось ничего, поэтому копирование работает корректно
```



```
while (l < r) //Копируем ответ из буфера обратно в массив a
    a[l] := buffer[l]
    l := l + 1
```

Факт подтвердился, но теперь ему стало интересно следующее — при данных n и k , существует ли массив длины n , на котором его сортировка будет делать ровно k сравнений? Более того, если такой массив существует, то Илье интересно посмотреть на любой пример такого массива. Сам он не смог придумать, как решать эту задачу, поэтому просит вас помочь ему.

Формат входных данных

Вам даны два целых числа, разделённые пробелом — n и k ($1 \leq n \leq 10^5; 0 \leq k \leq 10^9$)

Формат выходных данных

Если массива длины n , при сортировке которого `merge_sort` использует не более k сравнений, не существует, то выведите единственную строчку «No» (без кавычек).

Если же такой массив существует, то в первой строчке выведите слово «Yes» (без кавычек), а во второй строчке сам массив — n чисел, разделённых пробелами. Каждое число по модулю должно не превосходить 10^9 . Гарантируется, что если подходящий массив существует, то существует и подходящий массив с данным ограничением.

Примеры

стандартный ввод	стандартный вывод
4 5	YES 1 3 2 4
4 0	NO

Задача G. Прогульщик Петя

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

Петя учится в школе. Его класс занимается пять дней в неделю, однако каждый день у него шесть уроков. Пете кажется, что их у него слишком много, поэтому некоторые он решил прогулять. К сожалению, если по одному предмету он прогуляет больше одного урока, то преподаватели что-то заподозрят. Петру интересно, какое наибольшее количество уроков он сможет прогулять, не вызвав подозрений? Посчитайте это число для него.

Формат входных данных

Вам даны 30 строчек, разделённые на пять блоков по шесть строчек в каждой — петино расписание. В i -й строчке каждого блока записана строка, состоящая из строчных букв латинского алфавита — название предмета, который в данный день идёт i -м. Каждое название не превосходит по длине 15.

Формат выходных данных

В единственную строчку напишите ответ на задачу — максимальное количество уроков, которое сможет прогулять Петя.

Пример

стандартный ввод	стандартный вывод
algebra algebra geometry geometry biology topology philosophy algebra biology topology geography algebra literature chemistry philosophy algebra chemistry physics algebra topology geometry chemistry chemistry informatics informatics literature literature biology informatics algebra	10

Замечание

В тесте из условия Петя может прогулять десять уроков следующим образом: второй, третий и шестой уроки в понедельник; первый и пятый уроки во вторник; второй и шестой уроки в среду; последний урок в четверг, третий и четвёртый уроки в пятницу.

Задача Н. Александр учит стихи

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

Александр считает себя поэтом. Более того - он даже пишет стихи! Недавно у него появилось желание выучить своё самое красивое стихотворение. Для этого он хочет использовать новую технику запоминания стихов. Суть её заключается в том, чтобы взять от каждой строчки некоторый, возможно, пустой префикс, затем склеить полученные префиксы в одну строку не меняя порядка и уже её учить. К сожалению, Александр может запомнить только строчку длины k . Также он является эстетом и поэтому хочет, чтобы полученная строка была лексикографически минимальна. Помогите ему в нахождении такой строки.

Префиксом длины l строки $S = S_0S_1 \dots S_{n-1}$ называется строка $S_0S_1 \dots S_{l-1}$, либо пустая строка, если $l = 0$.

Строка S лексикографически меньше строки S^1 , если существует l , такое что префиксы длины l у обеих строк совпадают, а $S_l < S^1_l$, либо если S является префиксом строки S^1 , но не совпадает с S^1 .

Формат входных данных

В первой строке входных данных находятся два целых числа — n и k ($1 \leq n \leq 5 \cdot 10^3; 1 \leq k \leq 5 \cdot 10^3$) — количество строк в произведении Александра и длина искомой строки соответственно. В следующих n строках содержатся строчки из стихотворения Александра. Каждая строчка суть слово из строчных букв латинского алфавита. Гарантируется, что суммарная длина всех слов не превосходит $5 \cdot 10^3$, и что ответ существует.

Формат выходных данных

Выведите единственную строку - ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 7 aaaaa bbbbbb aaaaa	aaaaaaa
3 4 ab abb bba	aabb
5 5 are you can do it	acadi

Замечание

В первом тестовом примере можно взять первые 4 буквы из первой строки и первые 3 из третьей.

Во втором тесте можно взять по первой букве из первой и второй строки, а из третьей взять первые два символа.

В третьем тесте следует взять первый символ из первой строки, первые два символа из третьей и по одному первому символу из двух последних строк.