Задача A. Алгоритм Флойда

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.25 секунды
Ограничение по памяти:	1024 мебибайта

Рассмотрим неориентированный взвешенный полный граф из n вершин. Граф генерируется следующим алгоритмом:

```
state := s
for i := 0, ..., n-2:
  for j := i+1, ..., n-1:
    state := (state * 69’069 + 1) % 4’294’967’296
    cur := state % 1’000’000’000
    a[i][j] := cur
    a[j][i] := cur
```

Здесь `s` — начальное состояние для генератора псевдослучайных чисел, а `a[i][j]` — длина ребра из вершины `i` в вершину `j`. Найдите сумму кратчайших расстояний между всеми парами вершин.

Формат входных данных

В первой строке заданы два целых числа, `n` и `s` ($2 \leqslant n \leqslant 1024$; $1 \leqslant s \leqslant 10^9$).

Формат выходных данных

Выведите одно целое число: сумму кратчайших расстояний между всеми парами вершин. Пары (i, j) и (j, i) считаются одинаковыми.

Пример

стандартный ввод	***стандартный вывод***
2 1	69070

Задача B. Количество путей

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	1024 мебибайта

Рассмотрим поле из $n \times n$ клеток. Некоторые клетки свободны, в других находятся препятствия.

Заданы числа $a_1, a_2, \ldots, a_{n-1}$. Из клетки (r, c) в клетку $(r + k, c)$, если обе они свободны, ведут ровно a_k рёбер. Из клетки (r, c) в клетку $(r, c + k)$, если обе они свободны, также ведут ровно a_k рёбер.

Найдите количество путей по рёбрам из верхней левой клетки в правую нижнюю. В этих двух клетках препятствий нет. Поскольку ответ может быть очень большим, вычислите остаток от деления количества путей на $10^9 + 9$.

Формат входных данных

В первой строке задано целое число n ($2 \leqslant n \leqslant 1024$).

Во второй строке заданы $n - 1$ целых чисел $a_1, a_2, \ldots, a_{n-1}$ ($0 \leqslant a_i \leqslant 1000$).

Следующие n строк содержат по n символов и описывают поле. Решётка («`#`», ASCII-код 35) соответствует препятствиям, точка («`.`», ASCII-код 46) — свободным клеткам. Левая верхняя и правая нижняя клетки свободны.

Формат выходных данных

Пусть s — количество путей по рёбрам из левой верхней клетки в правую нижнюю. Выведите одно целое число: $s \bmod (10^9 + 9)$.

Примеры

стандартный ввод	**стандартный вывод**

`3`
`2 1`
`...`
`###`
`#..`
	`9`

`5`
`1000 999 454 961`
`.....`
`.....`
`.....`
`.##..`
`#....`
	`507366651`

Задача C. Коммивояжёры

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1.5 секунды
Ограничение по памяти:	1024 мебибайта

Есть n городов. Из каждого города в каждый есть ровно k односторонних дорог k разных цветов.

А ещё есть k коммивояжёров — по одному коммивояжёру каждого из k цветов. Каждый коммивояжёр хочет начать путешествие в каком-то городе и объехать города по дорогам своего цвета, побывав в каждом городе ровно один раз.

Для каждого из k коммивояжёров найдите длину кратчайшего пути.

Формат входных данных

В первой строке заданы целые числа n и k ($2 \leqslant n \leqslant 20$; $1 \leqslant k \leqslant 8$).

Далее следуют k описаний дорог: сначала все дороги первого цвета, затем все дороги второго цвета, и так далее. Каждое описание состоит из n строк, по n чисел в каждой: в u-й строке v-е число задаёт длину дороги из города u в город v. Длина каждой дороги — целое число от 1 до 10^8 при $u \neq v$ и 0 при $u = v$.

Формат выходных данных

Выведите k целых чисел: длину кратчайшего пути для каждого из k коммивояжёров.

Пример

стандартный ввод	**стандартный вывод**
3 2 0 1 2 1 0 2 1 2 0 0 3 2 4 0 1 5 6 0	2 4

Задача D. Десятичные цифры

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	1024 мебибайта

Даны целые числа n, s и k.

Выполним n раз присваивание $s \leftarrow (s \cdot 69\,069 + 1) \bmod 2^{32}$. После каждого присваивания прибавим к ответу число, составленное из k последних десятичных цифр s.

Посчитайте ответ.

Формат входных данных

В первой строке заданы три целых числа: n, s и k ($1 \leqslant n, s \leqslant 10^9$; $1 \leqslant k \leqslant 9$).

Формат выходных данных

Выведите ответ.

Примеры

стандартный ввод	*стандартный вывод*
`1 1 9`	`69070`
`3 12345 8`	`132437497`

Пояснения к примерам

Во втором примере:

- $s = 852\,656\,806$, последние 8 цифр равны $52\,656\,806$;
- $s = 3\,856\,338\,159$, последние 8 цифр равны $56\,338\,159$;
- $s = 1\,023\,442\,532$, последние 8 цифр равны $23\,442\,532$.

Задача E. Умножение-2

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.25 секунды
Ограничение по памяти:	1024 мебибайта

Найдите произведение двух заданных чисел.

Формат входных данных

Во входных данных заданы два целых неотрицательных числа A и B ($A \leqslant 10^{1\,000\,000}$; $B \leqslant 1000$), каждое в отдельной строке.

Формат выходных данных

Выведите одно число, равное произведению A и B.

Пример

стандартный ввод	**стандартный вывод**
5 3	15

Задача F. Множества: размер

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мебибайта

Рассмотрим множества, состоящие из целых чисел $1, 2, \ldots, m$.

Изначально заданы n таких множеств: $A_1, A_2, \ldots, A_n$. Далее q множеств получаются из предыдущих одной из четырёх операций:

- $A_k = A_i \cup A_j$ (объединение),
- $A_k = A_i \cap A_j$ (пересечение),
- $A_k = A_i \setminus A_j$ (разность),
- $A_k = A_i \,\Delta\, A_j$ (симметрическая разность).

Множество A_k ($n+1 \leqslant k \leqslant n+q$) может строиться из любых двух множеств A_i и A_j, определённых к этому моменту ($1 \leqslant i, j < k$).

Для каждого вновь построенного множества выведите количество элементов в нём.

Формат входных данных

В первой строке заданы целые числа m, n и q ($1 \leqslant m, n, q \leqslant 50\,000$).

В следующих n строках заданы изначальные множества $A_1, A_2, \ldots, A_n$. Каждая строка имеет вид «$\ell \;\; e_1 \; e_2 \; \ldots \; e_\ell$». Здесь ℓ — количество элементов во множестве ($1 \leqslant \ell \leqslant m$), а e_i — сами элементы в порядке возрастания ($1 \leqslant e_1 < e_2 < \ldots < e_\ell \leqslant m$). Общее количество элементов во всех изначальных множествах не превосходит $100\,000$.

Следующие q строк задают новые множества $A_{n+1}, A_{n+2}, \ldots, A_{n+q}$. Строка, задающая A_k, имеет вид «$i \;\; \xi \;\; j$». Здесь i и j — номера множеств-аргументов ($1 \leqslant i, j < k$), а ξ — обозначение операции: «`|`» для объединения, «`&`» для пересечения, «`\`» для разности и «`^`» для симметрической разности.

Формат выходных данных

Для каждого нового множества выведите одно целое число: количество элементов в нём.

Пример

стандартный ввод	стандартный вывод	пояснение
`9 3 4`	`1`	`n = 3,  q = 4`
`3  1 2 3`	`3`	`A1 = {1, 2, 3}`
`4  3 5 6 8`	`4`	`A2 = {3, 5, 6, 8}`
`2  1 9`	`3`	`A3 = {1, 9}`
`1 & 3`		`A4 = {1}`
`1 ^ 3`		`A5 = {2, 3, 9}`
`1 | 3`		`A6 = {1, 2, 3, 9}`
`6 \ 2`		`A7 = {1, 2, 9}`

Задача G. Множества: начало

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мебибайта

Рассмотрим множества, состоящие из целых чисел $1, 2, \ldots, m$.

Изначально заданы n таких множеств: $A_1, A_2, \ldots, A_n$. Далее q множеств получаются из предыдущих одной из четырёх операций:

- $A_k = A_i \cup A_j$ (объединение),
- $A_k = A_i \cap A_j$ (пересечение),
- $A_k = A_i \setminus A_j$ (разность),
- $A_k = A_i \,\Delta\, A_j$ (симметрическая разность).

Множество A_k ($n+1 \leqslant k \leqslant n+q$) может строиться из любых двух множеств A_i и A_j, определённых к этому моменту ($1 \leqslant i, j < k$).

Для каждого вновь построенного множества выведите его начало: минимальные три элемента по порядку. Если элементов меньше трёх, выведите их все, если больше — выведите после них три точки.

Формат входных данных

В первой строке заданы целые числа m, n и q ($1 \leqslant m, n, q \leqslant 50\,000$).

В следующих n строках заданы изначальные множества $A_1, A_2, \ldots, A_n$. Каждая строка имеет вид «$\ell \quad e_1 \;\; e_2 \;\; \ldots \;\; e_\ell$». Здесь ℓ — количество элементов во множестве ($1 \leqslant \ell \leqslant m$), а e_i — сами элементы в порядке возрастания ($1 \leqslant e_1 < e_2 < \ldots < e_\ell \leqslant m$). Общее количество элементов во всех изначальных множествах не превосходит $100\,000$.

Следующие q строк задают новые множества $A_{n+1}, A_{n+2}, \ldots, A_{n+q}$. Строка, задающая A_k, имеет вид «$i \;\; \xi \;\; j$». Здесь i и j — номера множеств-аргументов ($1 \leqslant i, j < k$), а ξ — обозначение операции: «|» для объединения, «&» для пересечения, «\» для разности и «^» для симметрической разности.

Формат выходных данных

Для каждого нового множества выведите его начало в отдельной строке.

Пример

стандартный ввод	стандартный вывод	пояснение
9 3 4 3 1 2 3 4 3 5 6 8 2 1 9 1 & 3 1 ^ 3 1 | 3 6 \ 2	1 2 3 9 1 2 3 ... 1 2 9	n = 3, q = 4 A1 = {1, 2, 3} A2 = {3, 5, 6, 8} A3 = {1, 9} A4 = {1} A5 = {2, 3, 9} A6 = {1, 2, 3, 9} A7 = {1, 2, 9}

Задача H. Суммы двух

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	1024 мебибайта

Фокусница Лина утверждает, что обычный современный компьютер легко может производить сотню миллиардов операций в секунду! Чтобы доказать это, она предлагает проделать следующие вычисления.

Пусть V — множество целых чисел, изначально пустое. Дано начальное значение числа s. Сделаем n шагов следующего вида:

- $s \leftarrow (s \cdot 618\,023 + 1) \bmod 999\,983$;
- найдём количество различных пар чисел из V, которые в сумме дают s;
- если это количество пар чётно, добавим число s в множество V.

Сколько элементов будет в множестве V после всех n шагов?

Формально: на каждом шаге мы считаем количество пар (a, b), в которых $a \in V$, $b \in V$, $a \leqslant b$ и $a + b = s$.

Формат входных данных

В первой строке заданы целые числа n и s ($1 \leqslant n \leqslant 200\,000$; $0 \leqslant s < 999\,983$; $s \neq 742\,681$).

Формат выходных данных

Выведите одно целое число: размер множества V после n шагов.

Пример

стандартный ввод	***стандартный вывод***
`4 179629`	`3`

Пояснение к примеру

В примере значения s на четырёх шагах равны $740\,740$, $139\,655$, $469\,353$ и $880\,395$.

Задача I. Разворот битов

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	1024 мебибайта

Даны целое число n, а также целые неотрицательные 32-битные числа s, a и c.

Заведём переменную для ответа: $answer \leftarrow 0$.

Выполним n раз следующие шаги:

- $s \leftarrow (s \cdot a + c) \bmod 2^{32}$;
- $s \leftarrow rev(s)$;
- $answer \leftarrow answer + s$.

Здесь $rev(s)$ — разворот битов в двоичном представлении числа s: бит 0 меняется местами с битом 31, …, бит 15 меняется местами с битом 16.

Посчитайте значение ответа $answer$ после n шагов.

Формат входных данных

В первой строке заданы четыре целых числа: n, s, a и c ($1 \leqslant n \leqslant 10^8$; $0 \leqslant s, a, c < 2^{32}$).

Формат выходных данных

Выведите ответ.

Пример

стандартный ввод	**стандартный вывод**
1 3987654321 2234567918 277	3315826862

Задача J. Сумма простых битов

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	1024 мебибайта

Даны целое число n, а также целые неотрицательные 32-битные числа s, a и c.

Выполним n раз присваивание $s \leftarrow (s \cdot a + c) \bmod 2^{32}$. После каждого присваивания прибавим к ответу количество единичных битов на простых позициях (2, 3, 5, ..., 31) в s.

Посчитайте ответ.

Формат входных данных

В первой строке заданы четыре целых числа: n, s, a и c ($1 \leqslant n \leqslant 10^9$; $0 \leqslant s, a, c < 2^{32}$).

Формат выходных данных

Выведите ответ.

Пример

стандартный ввод	**стандартный вывод**
`1 3987654321 2234567918 277`	`3`

Пояснение к примеру

Вот запись s после первого присваивания:

$$\underline{0}1\underline{1}10101\underline{0}000\underline{0}0\underline{0}111\underline{0}0\underline{0}101\underline{1}0\underline{1}0\underline{0}\underline{0}11$$

Подчёркиванием отмечены биты на простых позициях.