

Задача А. Алгоритм Флойда

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

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 ($2 \leq n \leq 1024$; $1 \leq s \leq 10^9$).

Формат выходных данных

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

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1	69070

Задача В. Количество путей

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

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

Заданы числа a_1, a_2, \dots, 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 ($2 \leq n \leq 1024$).

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

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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 ... ### #..	9
5 1000 999 454 961##.. #....	507366651

Задача С. Коммивояжёры

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

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 ($2 \leq n \leq 20$; $1 \leq k \leq 8$).

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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2	2
0 1 2	4
1 0 2	
1 2 0	
0 3 2	
4 0 1	
5 6 0	

Задача D. Десятичные цифры

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

Даны целые числа n , s и k .

Выполним n раз присваивание $s \leftarrow (s \cdot 69069 + 1) \bmod 2^{32}$. 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 ($1 \leq n, s \leq 10^9$; $1 \leq k \leq 9$).

Формат выходных данных

Выведите ответ.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
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 Е. Сумма вещественных

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

Найдите сумму n случайных вещественных чисел из $[0; 1)$.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 2 \cdot 10^6$).

Во второй строке заданы n вещественных чисел a_1, a_2, \dots, a_n ($0 \leq a_i < 1$). Каждое вещественное число выбрано случайно, равномерно и независимо от остальных в полуинтервале $[0; 1)$. Числа даны ровно с 10 знаками после десятичной точки.

Формат выходных данных

Выведите сумму n заданных чисел. Ответ считается верным, если относительная погрешность не превосходит 10^{-12} .

Пример

<i>стандартный ввод</i>
3 0.2965666396 0.2741007666 0.6664403653
<i>стандартный вывод</i>
1.2371077715

Задача F. Множества: размер

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

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

Изначально заданы n таких множеств: A_1, A_2, \dots, 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 ($n+1 \leq k \leq n+q$) может строиться из любых двух множеств A_i и A_j , определённых к этому моменту ($1 \leq i, j < k$).

Для каждого вновь построенного множества выведите количество элементов в нём.

Формат входных данных

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

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

Следующие q строк задают новые множества $A_{n+1}, A_{n+2}, \dots, A_{n+q}$. Строка, задающая A_k , имеет вид « $i \ \xi \ j$ ». Здесь i и j — номера множеств-аргументов ($1 \leq i, j < k$), 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

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
9 3 4	1	$n = 3, \ q = 4$
3 1 2 3	3	$A_1 = \{1, 2, 3\}$
4 3 5 6 8	4	$A_2 = \{3, 5, 6, 8\}$
2 1 9	3	$A_3 = \{1, 9\}$
1 & 3		$A_4 = \{1\}$
1 ^ 3		$A_5 = \{2, 3, 9\}$
1 3		$A_6 = \{1, 2, 3, 9\}$
6 \ 2		$A_7 = \{1, 2, 9\}$

Задача G. Множества: начало

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

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

Изначально заданы n таких множеств: A_1, A_2, \dots, 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 ($n+1 \leq k \leq n+q$) может строиться из любых двух множеств A_i и A_j , определённых к этому моменту ($1 \leq i, j < k$).

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 ($1 \leq m, n, q \leq 50\,000$).

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

Следующие q строк задают новые множества $A_{n+1}, A_{n+2}, \dots, A_{n+q}$. Строка, задающая A_k , имеет вид « $i \ \xi \ j$ ». Здесь i и j — номера множеств-аргументов ($1 \leq i, j < k$), 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

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
9 3 4	1	$n = 3, \ q = 4$
3 1 2 3	2 3 9	$A_1 = \{1, 2, 3\}$
4 3 5 6 8	1 2 3 ...	$A_2 = \{3, 5, 6, 8\}$
2 1 9	1 2 9	$A_3 = \{1, 9\}$
1 & 3		$A_4 = \{1\}$
1 ^ 3		$A_5 = \{2, 3, 9\}$
1 3		$A_6 = \{1, 2, 3, 9\}$
6 \ 2		$A_7 = \{1, 2, 9\}$

Задача Н. Множества: медиана

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

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

Изначально заданы n таких множеств: A_1, A_2, \dots, 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 ($n+1 \leq k \leq n+q$) может строиться из любых двух множеств A_i и A_j , определённых к этому моменту ($1 \leq i, j < k$).

Для каждого вновь построенного множества выведите его нижнюю медиану: элемент с номером $\lceil s/2 \rceil$, считая с единицы, если s — размер множества.

Формат входных данных

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

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

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

Формат выходных данных

Для каждого нового множества выведите его нижнюю медиану. Если множество пустое, выведите -1 .

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
9 3 4	1	$n = 3, \ q = 4$
3 1 2 3	3	$A_1 = \{1, 2, 3\}$
4 3 5 6 8	2	$A_2 = \{3, 5, 6, 8\}$
2 1 9	2	$A_3 = \{1, 9\}$
1 & 3		$A_4 = \{1\}$
1 ^ 3		$A_5 = \{2, 3, 9\}$
1 3		$A_6 = \{1, 2, 3, 9\}$
6 \ 2		$A_7 = \{1, 2, 9\}$

Задача I. Разворот битов

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

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 ($1 \leq n \leq 10^8$; $0 \leq s, a, c < 2^{32}$).

Формат выходных данных

Выведите ответ.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 3987654321 2234567918 277	3315826862

Задача J. Перемешивание битов

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

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:

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

Здесь $shuf(s)$ — следующее перемешивание битов в двоичном представлении числа s : сначала на места $0, 1, \dots, 10$ по порядку ставятся биты с простых позиций $2, 3, 5, \dots, 31$, а затем на места 11 и далее по порядку ставятся биты с не простых позиций $0, 1, 4, \dots, 30$.

Посчитайте значение ответа $answer$ после n шагов.

Формат входных данных

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

Формат выходных данных

Выведите ответ.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 3987654321 2234567918 277	3560610316

Пояснение к примеру

Вот запись s после первого присваивания:

01110101000000011100010110100011

А вот — после $shuf$:

11010100001110101001101000001100

Подчёркиванием отмечены биты на простых позициях и места, куда они перешли.

Задача К. Сумма простых битов

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

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 ($1 \leq n \leq 10^9$; $0 \leq s, a, c < 2^{32}$).

Формат выходных данных

Выведите ответ.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 3987654321 2234567918 277	3

Пояснение к примеру

Вот запись s после первого присваивания:

011101010100000011100010110100011

Подчёркиванием отмечены биты на простых позициях.