

Задача А. Могучие Промежуточники

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

Могучие Промежуточники выписали все степени числа k в k -ичной системе счисления подряд без пробелов. Сначала первую степень, затем вторую степень, третью степень и так далее до бесконечности. Дабы быть уверенными в верности своего могучего разума, побеждающего вселенских врагов этого мира, они решили проверить верность записи n -й цифры данного числа, узнать которую они вас и просят.

Формат входных данных

Вам даны два целых числа, порождённых разумом Могучих Промежуточников: k и n ($2 \leq k \leq 36$, $1 \leq n \leq 10^{18}$) — основание системы счисления и номер позиции.

Формат выходных данных

Выведите n -ю цифру получившегося числа. В системе счисления по основанию k в качестве цифр используются первые k символов строки `0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ` в заданном порядке.

Система оценки

В задаче 4 подзадачи. Подзадача 0 — тесты из условия, за неё баллы не начисляются. Тестирование подзадачи начинается, если пройдены все тесты в необходимых подзадачах. Система оценки «полная» означает, что решению будут начисляться баллы только при успешном прохождении всех тестов данной подзадачи.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Система оценки
0	0	Тесты из условия	—	—
1	41	$N \leq 10^6$	0	полная
2	32	$N \leq 10^{12}$	0, 1	полная
3	27	—	0, 1, 2	полная

Пример

стандартный ввод	стандартный вывод
2 1	1

Замечание

Следует обратить внимание, что число n в этой задаче не помещается в стандартный 32-битный тип данных. Необходимо использовать 64-битный тип данных (`long long` в C++, `int64` в Паскале, `long` в Java).

Задача В. Карточная игра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Имеется набор карточек для игры. На каждой карточке написано число, причём числа на разных карточках могут совпадать. В игру играют два человека. В каждом раунде игроки вытаскивают наугад по одной карточке и оставляют её у себя. У кого число на карточке оказалось больше, тот и победил в этом раунде. А если числа совпали, то в этом раунде объявляется ничья. Игра продолжается, пока не закончатся все карточки.

Нельзя заранее предсказать, сколько за время игры будет «ничьих» и побед у каждого игрока. Если на карточках были написаны числа 1, 1, 2, 3, 4, 5, то можно, например, сыграть 3 таких раунда: (2, 5), (4, 3), (1, 1). В этом случае первый раунд выиграл второй игрок, второй раунд выиграл первый игрок, а третий раунд закончился «ничьей».

Спрашивается, а сколько существует разных *интересных* вариантов сыграть состоящую из N раундов игру, если известны числа, написанные на карточках? Вариант считается *интересным*, если количество «ничьих» максимально для данного набора карточек, а также первый игрок выиграл ровно K раз. Два варианта, отличающиеся только порядком раундов, считаются одинаковыми. Карточки, на которых записаны одинаковые числа, не отличаются друг от друга.

Формат входных данных

В первой строке на вход подается N — количество раундов игры ($2 \leq N \leq 50\,000$) и K — количество побед первого игрока ($0 \leq K \leq N$).

Во второй строке записаны $2N$ натуральных чисел, не превосходящих 10^9 . Эти числа могут повторяться, они написаны на карточках.

Формат выходных данных

На выходе должно быть записано количество различных вариантов сыграть N игр так, чтобы количество «ничьих» было максимально возможным, а число побед первого игрока было равно K . Искомое число вариантов вычислить по модулю $10^9 + 7$. Порядок раундов в игре не важен. Важно лишь, карточки с какими числами участвовали в каждом раунде.

Система оценки

В задаче 4 подзадачи. Подзадача 0 — тесты из условия, за неё баллы не начисляются. Тестирование подзадачи начинается, если пройдены все тесты в необходимых подзадачах. Система оценки «полная» означает, что решению будут начисляться баллы только при успешном прохождении всех тестов данной подзадачи.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Система оценки
0	0	Тесты из условия	—	—
1	14	$N \leq 5$	0	полная
2	45	$N \leq 1000$	0, 1	полная
3	24	$K \leq 100$	0, 1	полная
4	17	—	0 – 3	полная

Примеры

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

Замечание

Пояснения к первому примеру: нужно сыграть 3 раунда. Максимальное количество «ничьих» при заданных числах на карточках равно 1. Чтобы в двух оставшихся раундах выиграл первый игрок, возможны всего 3 варианта сыграть эти три раунда: $[(6, 5), (4, 2), (4, 4)]$, $[(6, 4), (5, 2), (4, 4)]$, $[(6, 2), (5, 4), (4, 4)]$.

Пояснения ко второму примеру: нужно сыграть 2 раунда. Максимальное количество «ничьих» при заданных числах на карточках равно 1, поэтому первый игрок не может выиграть два раза.

Задача С. Грибочки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

О-о, опушечка
О-о-о, подосиновичек
О-о-о, рыжик...

— «Грибочки»

В лесу неподалёку растёт очень много грибов и находится ещё большее количество опушечек. Для удобства ориентации в лесу грибники пронумеровали опушечки леса номерами от 1 до n . Также известно, что вход в лес находится на опушечке под номером s .

Грибники протоптали в лесу целую сеть тропинок. Любая тропинка соединяет между собой две опушечки в лесу. На каждой опушечке либо растёт ровно один гриб, либо не растёт ничего. Чтобы не теряться в лесу, грибники проложили лишь самые необходимые тропинки, поэтому между любыми двумя опушечками в лесу существует только одна связывающая их последовательность тропинок, в которой тропинки не повторяются. Кроме того, известно, что грибы растут на тех и только тех опушечках, которые не являются промежуточными ни на одном пути от входа в лес до другой опушечки.

Вы хотите сходить в лес и собрать в нём как можно больше грибов. Начиная от входа в лес, вы можете ходить по всем тропинкам сколько угодно раз. Однако, если вы не хотите заблудиться, то лучше запомнить самую первую тропинку, по которой вы прошли, и не ходить по ней больше одного раза. Какое максимальное количество грибов вы можете собрать?

Формат входных данных

В первой строке входного файла записаны два целых числа n и s — количество опушечек и номер опушечки, на которой расположен вход в лес соответственно ($6 \leq n \leq 10^5$, $1 \leq s \leq n$).

В каждой из следующих $n - 1$ строк записано по два числа u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) — это номера опушечек, являющихся концами i -й тропинки.

Формат выходных данных

В единственной строке ваша программа должна вывести два целых числа. Первое число — это максимально возможное количество грибов, которое можно собрать. Второе число — это номер первой опушечки, на которую нужно пойти, чтобы собрать как можно больше грибов при условии, что вы не будете больше одного раза ходить по самой первой тропинке, по которой вы прошли. Если таких опушечек несколько, выведите любую из них.

Система оценки

В задаче 4 подзадачи. Подзадача 0 — тесты из условия, за неё баллы не начисляются. Тестирование подзадачи начинается, если пройдены все тесты в необходимых подзадачах. Система оценки «полная» означает, что решению будут начисляться баллы только при успешном прохождении всех тестов данной подзадачи.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Система оценки
0	0	Тесты из условия	—	—
1	10	$n \leq 100$	0	полная
2	40	$n \leq 1000$	0, 1	полная
3	50	—	0, 1, 2	полная

Пример

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

Задача D. Тройки на доске

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Девочка Аня выписала на доске N различных чисел в порядке возрастания, а затем задумалась, что с ними можно сделать интересного? Проходящий мимо мальчик Боря заметил раздумья Ани и предложил ей сыграть в следующую игру: надо выписать на доску все упорядоченные тройки чисел (a, b, c) такие, что квадратное уравнение $ax^2 + bx + c = 0$ ($a \neq 0$) имеет по крайней мере один вещественный корень. Две тройки (a_1, b_1, c_1) и (a_2, b_2, c_2) считаются различными, если $a_1 \neq a_2$ или $b_1 \neq b_2$ или $c_1 \neq c_2$. Одно и то же число можно использовать больше одного раза. Школьники выписали какое-то количество троек и решили проверить себя. Помогите им посчитать, сколько всего таких упорядоченных троек должно быть выписано на доске.

Формат входных данных

В первой строке записано натуральное число N ($2 \leq N \leq 6000$) — количество чисел, записанных на доске.

Во второй строке записано N натуральных чисел a_i ($0 \leq a_1 < a_2 < \dots < a_n \leq 10^9$) — список чисел, записанных на доске.

Формат выходных данных

В единственной строке выведите количество упорядоченных троек, которое нужно будет выписать на доске по модулю 274876858367.

Система оценки

В задаче 4 подзадачи. Подзадача 0 — тесты из условия, за неё баллы не начисляются. Тестирование подзадачи начинается, если пройдены все тесты в необходимых подзадачах. Система оценки «полная» означает, что решению будут начисляться баллы только при успешном прохождении всех тестов данной подзадачи.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Система оценки
0	0	Тесты из условия	—	—
1	21	$N \leq 500$	0	полная
2	22	$N \leq 1100$	0, 1	полная
3	31	$N \leq 2000$	0, 1, 2	полная
4	26	$N \leq 6000$	0, 1, 2, 3	полная

Пример

стандартный ввод	стандартный вывод
5 1 2 3 4 5	24

Задача Е. Родительские хромосомы

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

Хромосома в вольной интерпретации программиста – это символьная строка, в которой каждый отдельный символ – это ген. При скрещивании двух родительских хромосом одинаковой длины получается новая хромосома такой же длины. Каждый её ген достаётся «по наследству» случайным образом от одной из двух родительских хромосом. Причём первый ген хромосомы потомка – это первый ген какого-либо из родителей, второй ген хромосомы потомка – это второй ген какого-либо из родителей и т.д. Из-за фактора «случайности» у пары родительских хромосом может получиться много разных потомков. А можно ли, имея много разных родительских хромосом, установить – какие из них действительно являются родителями заданной хромосомы?

Формат входных данных

В первой строке на вход подается n – количество хромосом ($2 \leq n \leq 2 \cdot 10^4$) и l – длина хромосом ($2 \leq l \leq 50$). В каждой из следующих n строк записано по одной родительской хромосоме. Далее в строке записано натуральное число k – количество хромосом возможных потомков ($1 \leq k \leq 10$). Далее в каждой из k строк записано по одной хромосоме возможного потомка. Все родительские хромосомы и хромосомы возможных потомков – попарно различны и состоят из малых латинских букв.

Формат выходных данных

На выходе должны быть перечислены в возрастающем порядке номера тех потомков, чьи хромосомы можно получить при скрещивании двух родительских хромосом, набор которых задан во входном файле. Если ни для одной из хромосом потомков не нашлось пары родительских хромосом, выдать 0.

Система оценки

В задаче 4 подзадачи. Подзадача 0 – тесты из условия, за неё баллы не начисляются. Тестирование подзадачи начинается, если пройдены все тесты в необходимых подзадачах. Система оценки «полная» означает, что решению будут начисляться баллы только при успешном прохождении всех тестов данной подзадачи.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Система оценки
0	0	Тесты из условия	—	—
1	10	$n \leq 2$	0	полная
2	20	$n \leq 10^3$	0, 1	полная
3	40	$n \leq 10^4$	0, 1, 2	потестово
4	30	Дополнительных ограничений нет	0, 1, 2, 3	потестово

Примеры

стандартный ввод	стандартный вывод
4 6 abcbac bbcaab caccba bcacca 3 bacaba abcccb bcccab	1 3
2 2 st ss 1 tt	0