Задача A. Адаптивное избыточное кодирование

Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

В секретной лаборатории разрабатывается квантовый передатчик для передачи данных по зашумленному каналу связи. Особенностью технологии передачи является то, что при передаче возможны изменения переданных данных, но передающей стороне становится известно о произошедших изменениях, что позволяет скорректировать дальнейшую передачу.

Требуется передать массив битов, состоящий из N нулей и единиц. Для этого разрешается несколько раз осуществлять отправление блока данных. Каждый блок представляет собой массив из нулей и единиц. После отправления блока в него могут быть внесены ошибки. Каждая ошибка заключается в том, что некоторый бит блока заменяется на противоположный. После этого отправителю блока сообщается, что получит принимающая сторона.

При передаче разрешается отправить не более 100 блоков. При этом на суммарный размер отправленных блоков накладывается ограничение, различное для разных подзадач. Гарантируется, что суммарно в отправленные блоки будет внесено не более S ошибок.

Принимающей стороне передаётся конкатенация отправленных блоков со всеми внесёнными ошибками.

Вам требуется реализовать отправление и приём данных.

Точнее, вам нужно реализовать две процедуры: `transmit` и `receive`.

Процедура `transmit(int N, int S, int D[])` вызывается, чтобы передать массив битов. Её параметры:

- `N` — размер массива, $1000 \leqslant N \leqslant 100\,000$.
- `S` — суммарное число ошибок, которые могут появиться в отправленных блоках, $1 \leqslant S \leqslant 100$.
- `D` — массив, который требуется передать, массив имеет длину N, каждый его элемент равен 0 или 1.

При своей работе процедура `transmit` должна для отправки блока вызывать процедуру `send(int L, int F[], int R[])`. Её параметры:

- `L` — размер передаваемого блока, $1 \leqslant L \leqslant 100\,000$.
- `F` — массив, содержащий отправляемое сообщение, массив должен иметь длину L, каждый его элемент должен быть равен 0 или 1.
- `R` — массив, который после вызова функции будет содержать полученное сообщение, массив имеет длину L, каждый его элемент равен 0 или 1.

При кодировании сообщения процедура `send` должна быть вызвана не более 100 раз.

Процедура `receive(int N, int S, int L, int D[])` вызывается, чтобы получить сообщение. Её параметры:

- `N` — размер передаваемого массива, $1000 \leqslant N \leqslant 100\,000$.
- `S` — суммарное число ошибок, которые могли быть допущены при передаче, $1 \leqslant S \leqslant 100$.
- `L` — размер полученного сообщения, $1 \leqslant L \leqslant 1\,000\,000$.
- `D` — массив, содержащий полученное сообщение, массив имеет длину L, каждый его элемент равен 0 или 1.

Декодировав сообщение, `receive` должна вызывать процедуру `answer(int V[])`. Её параметры:

- `V` — массив, содержащий переданное сообщение, массив должен иметь длину N, каждый его элемент должен быть равен 0 или 1.

Гарантируется, что на вход процедуре `receive` будут подаваться только блоки, сформированные процедурой `transmit`.

Обратите внимание, что, в отличие от выданного вам примера грейдера, при окончательном тестировании между вызовами `transmit` и `receive` ваше решение будет перезапущено. Поэтому бессмысленно пытаться обмануть тестирующую систему и сохранить информацию о передаваемом сообщении в памяти.

Выданный вам пример грейдера получает данные в следующем формате: длина сообщения, биты сообщения через пробел, количество ошибок, позиции, в которых произойдут ошибки.

Пример

В описании примера при передаче параметров массивы обозначаются как список элементов, заключённый в фигурные скобки. При написании программы соблюдайте синтаксис своего языка.

Пусть требуется передать массив `{0, 1, 0, 0, 1, 1}`, причём при передаче может быть внесено не более 2 ошибок.

Процедура `transmit` будет вызвана с параметрами `transmit(6, 2, {0, 1, 0, 0, 1, 1}`. Она, например, может вызывать процедуру `send` 6 раз, i-й раз передавая строку из 10 копий передаваемого бита. Таким образом, например, первый вызов будет таким: `send(10, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, R)`, где массив `R` должен иметь размер не менее 10. Пусть, например, при передаче этого массива вносится одна ошибка в 4-й бит. Тогда после вызова массив `R` будет содержать `{0, 0, 0, 1, 0, 0, 0, 0, 0, 0}`. Пусть, например, вторая ошибка вносится в последний 10-й бит 6-го вызова. Тогда после этого вызова `send(10, {1, 1, 1, 1, 1, 1, 1, 1, 1, 1}, R)` массив `R` будет содержать `{1, 1, 1, 1, 1, 1, 1, 1, 1, 0}`.

Для раскодирования переданного сообщения процедура `receive` получит следующие параметры: `receive(6, 2, 60, D)`, где массив `D` содержит все отправленные блоки с учётом ошибок: `{0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0}`.

Декодировав сообщение, процедура `receive` должна вызвать процедуру `answer({0, 1, 0, 0, 1, 1})`.

Система оценки

Для оценки используется несколько подзадач. Баллы за подзадачу начисляются, только если все тесты в этой подзадаче пройдены.

Тест считается пройденным, если при кодировании не были превышены ограничения, а при декодировании было корректно восстановлено исходное сообщение.

Подзадача 1

10 баллов. $S = 1$. Суммарный размер всех отправленных сообщений при кодировании не должен превышать $3N$.

Подзадача 2

15 баллов. $S = 1$. Суммарный размер всех отправленных сообщений при кодировании не должен превышать $1.1N$.

Подзадача 3

15 баллов. $1 \leqslant S \leqslant 100$, $N = 1000$. Суммарный размер всех отправленных сообщений при кодировании не должен превышать 10^6.

Подзадача 4

60 баллов. $1 \leqslant S \leqslant 100$, $1000 \leqslant N \leqslant 100\,000$. Суммарный размер всех отправленных сообщений при кодировании не должен превышать $1.2N$

Оценка потестовая.

Задача B. RSA. Взлом RSA

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

В 1977 году Ronald Linn Rivest, Adi Shamir и Leonard Adleman предложили новую криптографическую схему RSA, используемую до сих пор. RSA является криптосистемой с открытым ключом: зашифровать сообщение может кто угодно, знающий общеизвестный открытый ключ, а расшифровать сообщение — только тот, кто знает специальный секретный ключ.

Желающий использовать систему RSA для получения сообщений должен сгенерировать два простых числа p и q, вычислить $n = pq$ и сгенерировать два числа e и d такие, что $ed \equiv 1 \pmod{(p-1)(q-1)}$ (заметим, что $(p-1)(q-1) = \varphi(n)$). Числа n и e составляют открытый ключ и являются общеизвестными. Число d является секретным ключом, также необходимо хранить в тайне и разложение числа n на простые множители, так как это позволяет вычислить секретный ключ d.

Сообщениями в системе RSA являются числа из $\mathbb{Z}_n$. Пусть M — исходное сообщение. Для его шифрования вычисляется значение $C = M^e \bmod n$ (для этого необходимо только знание открытого ключа). Полученное зашифрованное сообщение C передается по каналу связи. Для его расшифровки необходимо вычислить значение $M = C^d \bmod n$, а для этого необходимо знание секретного ключа.

Вы перехватили зашифрованное сообщение C и знаете только открытый ключ: числа n и e. “Взломайте” RSA — расшифруйте сообщение на основе только этих данных.

Формат входных данных

Программа получает на вход три натуральных числа: n, e, C, $n \leqslant 10^9$, $e \leqslant 10^9$, $C < n$. Числа n и e являются частью какой-то реальной схемы RSA, т.е. n является произведением двух простых и e взаимно просто с $\varphi(n)$. Число C является результатом шифрования некоторого сообщения M.

Формат выходных данных

Выведите одно число M $(0 \leqslant M < n)$, которое было зашифровано такой криптосхемой.

Примеры

стандартный ввод	стандартный вывод
143 113 41	123
9173503 3 4051753	111111

Задача C. RSA. Взлом RSA-2

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

Петины друзья Вася, Коля и Паша используют шифрование RSA с разными большими ключами n_a, n_b, n_c, а в качестве открытого ключа e все трое используют $e = 3$. Петя послал всем трём друзьям одинаковое сообщение, не добавляя соль.

Покажите Пете, что он неправ, и что его сообщение можно расшифровать если перехватить три зашифрованных сообщения c_a, c_b, c_c.

Формат входных данных

Во входном файле заданы числа n_a, c_a, n_b, c_b, n_c, c_c

Все числа не превосходят 10^{50}, ключи n_a, n_b и n_c попарно взаимнопросты.

Формат выходных данных

Выведите сообщение m

Примеры

стандартный ввод	стандартный вывод
15 11 119 22 143 44	11