

Амбулаторная 3. СНМ, ПСД и ТЧ

А. Парковка

2 секунды, 256 мегабайт

На кольцевой парковке есть n мест пронумерованных от 1 до n . Всего на парковку приезжает n машин в порядке нумерации. У i -й машины известно место p_i , которое она хочет занять. Если машина приезжает на парковку, а её место занято, то она едет далее по кругу и встаёт на первое свободное место.

Входные данные

В первой строке входного файла находится число n ($1 \leq n \leq 300\,000$) — размер парковки и число машин. Во второй строке записаны n чисел, i -е из которых p_i ($1 \leq p_i \leq n$) — место, которое хочет занять машина с номером i .

Выходные данные

Выведите n чисел: i -е число — номер парковочного места, которое было занято машиной с номером i .

входные данные
3 2 2 2
выходные данные
2 3 1

В. Подсчет опыта

2 секунды, 64 мегабайта

В очередной онлайн игре игроки, как обычно, сражаются с монстрами и набирают опыт. Для того, чтобы сразаться с монстрами, они объединяются в кланы. После победы над монстром, всем участникам клана, победившего его, добавляется одинаковое число единиц опыта. Особенностью этой игры является то, что кланы никогда не распадаются и из клана нельзя выйти. Единственная доступная операция — объединение двух кланов в один.

Поскольку игроков стало уже много, вам поручили написать систему учета текущего опыта игроков.

Входные данные

В первой строке входного файла содержатся числа n ($1 \leq n \leq 200\,000$) и m ($1 \leq m \leq 200\,000$) — число зарегистрированных игроков и число запросов.

В следующих m строках содержатся описания запросов. Запросы бывают трех типов:

- `join X Y` — объединить кланы, в которые входят игроки X и Y (если они уже в одном клане, то ничего не меняется).
- `add X V` — добавить V единиц опыта всем участникам клана, в который входит игрок x ($1 \leq V \leq 100$).
- `get X` — вывести текущий опыт игрока x .

Изначально у всех игроков 0 опыта и каждый из них состоит в клане, состоящим из него одного.

Выходные данные

Для каждого запроса `get X` выведите текущий опыт игрока X .

входные данные
3 6 add 1 100 join 1 3 add 1 50 get 1 get 2 get 3
выходные данные
150 0 50

С. Реструктуризация компании

2 секунды, 256 мегабайт

В жизни даже самой успешной компании может наступить кризисный период, когда приходится принимать тяжёлое решение о реструктуризации, распускать и объединять отделы, увольнять работников и заниматься прочими неприятными делами. Рассмотрим следующую модель компании.

В Большой Софтверной Компании работают n человек. Каждый человек принадлежит какому-то *отделу*. Исходно каждый человек работает над своим проектом в своём собственном отделе (таким образом, в начале компания состоит из n отделов по одному человеку).

Однако, в жизни компании наступили тяжёлые времена, и руководство было вынуждено нанять кризисного менеджера, который начал переустраивать рабочий процесс для повышения эффективности производства. Обозначим за $team(person)$ команду, в которой работает человек $person$. Кризисный менеджер может принимать решения двух типов:

- Объединить отделы $team(x)$ и $team(y)$, сформировав из них один большой отдел, содержащий всех сотрудников $team(x)$ и $team(y)$, где x и y ($1 \leq x, y \leq n$) — номера каких-то двух сотрудников компании. Если $team(x)$ совпадает с $team(y)$, ничего делать не требуется.
- Объединить отделы $team(x)$, $team(x + 1)$, ..., $team(y)$, где x и y ($1 \leq x \leq y \leq n$) — номера каких-то двух сотрудников компании.

При этом кризисный менеджер иногда может интересоваться, работают ли в одном отделе сотрудники x и y ($1 \leq x, y \leq n$).

Помогите кризисному менеджеру, ответив на все его запросы.

Входные данные

Первая строка входных данных содержит два целых числа n и q ($1 \leq n \leq 200\,000$, $1 \leq q \leq 500\,000$) — количество сотрудников компании и количество запросов кризисного менеджера.

В последующих q строках находятся запросы кризисного менеджера. Каждый запрос имеет вид $type\ x\ y$, где $type \in \{1, 2, 3\}$. Если $type = 1$ или $type = 2$, то запрос представляет собой решение кризисного менеджера об объединении отделов соответственно первого или второго вида. Если $type = 3$, то требуется определить, работают ли в одном отделе сотрудники x и y . Обратите внимааие, что x может равняться y в запросе любого типа.

Выходные данные

На каждый запрос типа 3 выведите «YES» или «NO» (без кавычек), в зависимости от того, работают ли в одном отделе соответствующие люди.

входные данные
8 6 3 2 5 1 2 5 3 2 5 2 4 7 2 1 2 3 1 7
выходные данные
NO YES YES

D. Персистентный стек

2.5 c⌚, 256 мегабайт

Реализуйте персистентный стек.

Входные данные

Первая строка содержит количество действий n ($1 \leq n \leq 200\,000$). В строке номер $i + 1$ содержится описание действия i :

- $t \ m$ — добавить в конец стека номер t ($0 \leq t < i$) число m ($0 < m \leq 1000$);
- $t \ 0$ — удалить последний элемент стека номер t ($0 \leq t < i$).
Гарантируется, что стек t не пустой.

В результате действия i , описанного в строке $i + 1$, создается стек номер i . Изначально имеется пустой стек с номером ноль. Все числа во входном файле целые.

Выходные данные

Для каждой операции удаления выведите удаленный элемент на отдельной строке.

входные данные
8 0 1 1 5 2 4 3 2 4 3 5 0 6 6 1 0
выходные данные
3 1

E. Наборщик-рак

3 секунды⌚, 256 мегабайт

Задачи - Codeforces

Некоторые люди говорят, что Леонардо был большим почитателем Иоганна Гутенберга, немецкого кузнеца, который изобрел подвижную (наборную) печать, и что он воздал должное, сконструировав машину, названную им наборщик-рак — *il gambero scrivano* — очень простое наборное устройство. Оно чем-то похоже на современную простую пишущую машинку и выполняет всего 2 команды: одна, чтобы напечатать следующий символ, и вторая, чтобы отменить несколько последних команд. Замечательным свойством наборщика-рака является исключительная мощность команды отмены, которая рассматривается сама по себе как команда и тоже может быть отменена.

Вам необходимо реализовать программную модель наборщика-рака: она начинает работу с пустого текста, обрабатывает последовательность команд, передаваемых ей, и запросы относительно определенных позиций в текущем состоянии текста, как описано ниже.

- Напечатать букву L . Добавляет в конец текста один символ L — маленькую букву из диапазона $a \dots z$.
- Отменить A команд. Отменяет последние A команд, где A — положительное целое число.
- Узнать символ на позиции P . Выводит символ — букву, находящуюся в позиции P текущего текста, где P — неотрицательное целое число. Первая буква текста имеет индекс 0. (Этот запрос не является командой и поэтому игнорируется командой отмены.)

Команда отмены отменяет предыдущие U команд в обратном порядке. Если отменяемая команда — это напечатать символ L , то из конца текста удаляется буква L . Если отменяемая команда — это отменить X команд, то для этого значения X она заново применяет предыдущие X команд в их оригинальном порядке.

Входные данные

В первой строке дано число n ($1 \leq n \leq 1\,000\,000$) — количество запросов.

В следующих n строках дано описание запросов. Запрос начинается с символа.

- Если этот символ равен T, то это запрос напечатать символ L , и далее в строке дан символ L .
- Если этот символ равен U, то это запрос отменить A команд, и далее в строке дано число A . Гарантируется, что A не будет превышать количество ранее полученных команд.
- Если этот символ равен P, то это запрос вывести символ на позиции P , и далее в строке дано число P . Символы в строке нумеруются с 0. Гарантируется, что P будет меньше чем текущая длина текста (количество букв в текущем тексте).

Выходные данные

На одной строке выведите ответы на все запросы третьего типа.

входные данные
14 T a T b P 1 T d U 2 U 1 P 2 T e U 1 U 5 T c P 2 U 2 P 2
выходные данные
bdcd

Пояснение к примеру:

Вызов	Результат	Текущий текст
T a		a
T b		ab
P 1	b	ab
T d		abd
U 2		a
U 1		abd
P 2	d	abd
T e		abde
U 1		abd
U 5		ab
T c		abc
P 2	c	abc
U 2		abd
P 2	d	abd

F. Массовая проверка простоты

1.5 секунд🕒, 256 мегабайт

Целое число $p \geq 2$ является простым, если у него нет делителей кроме 1 и p . Необходимо для всех чисел во входном файле проверить простые они или нет.

Входные данные

В первой строке задано число n ($2 \leq n \leq 500\,000$). В следующих n строках заданы числа a_i ($2 \leq a_i \leq 2 \cdot 10^7$), которые нужно проверить на простоту

Выходные данные

Для каждого числа во входном файле выведите на отдельной строке «YES» или «NO» в зависимости от того, простое оно или нет.

входные данные
4 60 14 3 55
выходные данные
NO NO YES NO

G. Больше простых!

10 секунд🕒, 64 мегабайта

Найдите все простые числа не большие n . Поскольку n в этой задаче не просто большое, а прямо здоровенное, для того чтобы проверить, что вы нашли числа правильно, мы попросим вас посчитать от найденных чисел специальный хеш.

Хеш будет считаться по следующему алгоритму. В начале переменная $h = 0$. После каждого очередного встреченного простого числа p_i , будем пересчитывать h по формуле $h = h \cdot x + p_i$, при этом будем игнорировать переполнение знакового 32-битного целого типа. Значение переменной h в конце — это хеш, который вам нужно вывести.

Входные данные

Входной файл содержит два числа n ($2 \leq n \leq 10^9$) и x ($1 \leq x \leq 10^9$).

Выходные данные

Выведите полученный хеш.

входные данные
10 10
выходные данные
2357

входные данные
11 100
выходные данные
203050711

входные данные
1000000000 2
выходные данные
1576840463

H. Разложение на множители

2 секунды🕒, 256 мегабайт

Дано число. Требуется разложить его на простые множители.

Входные данные

Вводится число n ($2 \leq n \leq 10^9$).

Выходные данные

Выведите через пробел разложение на простые множители в порядке неубывания множителей.

входные данные
17
выходные данные
17

входные данные
60
выходные данные
2 2 3 5

I. Массовое разложение на множители

2 секунды🕒, 256 мегабайт

Дано много чисел. Требуется разложить их все на простые множители.

Входные данные

В первой строке задано число n ($2 \leq n \leq 300000$). В следующих n строках заданы числа a_i ($2 \leq a_i \leq 2 \cdot 10^6$), которые нужно разложить на множители.

Выходные данные

Для каждого числа выведите в отдельной строке разложение на простые множители в порядке возрастания множителей.

входные данные
4 60 14 3 55
выходные данные
2 2 3 5 2 7 3 5 11

J. МеганОД

2 секунды🕒, 64 мегабайта

Найдите НОД n заданных чисел.

Входные данные

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество чисел. Во второй строке заданы n целых чисел, не превышающие по модулю 10^9 .

Выходные данные

Выведите НОД данных n чисел.

входные данные
2 90 35
выходные данные
5

K. НОД подмножества

2 секунды🕒, 256 мегабайт

Задачи - Codeforces

Сегодня на уроке математики шестиклассник Петя изучил понятие наибольшего общего делителя. Петя тут же решил применить полученные знания на практике.

Петя выписал на листке бумаги n чисел a_1, \dots, a_n — номера домов, в которых живут его друзья. Теперь он хочет выбрать такое подмножество этих чисел, чтобы их наибольший общий делитель был равен его любимому числу d .

Помогите Пете выбрать из выписанных чисел искомое подмножество.

Входные данные

Первая строка входного файла содержит два целых числа n и d ($1 \leq n \leq 1000, 1 \leq d \leq 10^9$). Вторая строка содержит n целых чисел: a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Выходные данные

Если существует искомое подмножество, выведите на первой строке выходного файла число k — количество чисел в нем. На второй строке выведите числа, входящие в это подмножество.

Если решения не существует, выведите на первой строке выходного файла число -1 .

Если возможных ответов несколько, выведите любой из них.

входные данные
4 3 6 8 12 9
выходные данные
3 6 12 9

входные данные
3 3 2 4 8
выходные данные
-1

L. НОД и НОК

2 секунды🕒, 64 мегабайта

Двум школьникам из параллели C преподаватель дал следующее простое задание: для двух чисел a и b требуется посчитать их наименьший общий делитель (НОД) и наибольшее общее кратное (НОК). Школьники хотели не выполнять задание, а играть в настольные игры, поэтому они попросили друзей из старшей параллели сделать его за них. Когда они незадолго до отбоя вернулись, бумажка с числами a и b куда-то исчезла, а без неё преподаватель не сможет проверить правильность выполнения задания. Помогите двум школьникам по известным значениям НОД и НОК исходных чисел найти все подходящие пары a и b .

Входные данные

В единственной строке содержатся два целых числа x и y ($1 \leq x \leq y \leq 10^9$). Здесь x — значение наибольшего общего делителя чисел a и b , а y — значение наименьшего общего кратного чисел a и b .

Гарантируется, что y делится на x .

Выходные данные

Выведите все пары чисел a и b , такие что $\text{НОД}(a, b) = x$ и $\text{НОК}(a, b) = y$, в порядке возрастания a . Если у нескольких пар значения a совпадают, то выведите эти пары в порядке возрастания b .

входные данные
1 15

выходные данные
1 15
3 5
5 3
15 1