

Лабораторная работа №3

В рамках данной лабораторной работы необходимо решить предложенные задачи на языке программирования высокого уровня из предложенного перечня. Варианты задач находятся в отдельном файле. В качестве результата выполнения практической работы необходимо приложить **архив с файлами решенных задач, а также отчет о выполнении работы**. Файлы решенных задач представляют собой файлы с исходным кодом с установленным расширением.

Отчет о выполнении практической работы должен содержать:

1. Титульный лист

2. Содержание

3. 4 параграфа, в которых раскрыто решение каждой задачи. По каждой задаче необходимо представить следующую информацию:

3.1. *Условие задачи*. Берется из файла.

3.2. *Ход решения задачи*. Студент описывает логику решения данной задачи. Какие алгоритмы и для чего использованы, как построена программа. Данная часть является описательной. Здесь следует говорить именно о построении алгоритма, опуская процессы ввода и вывода данных (если это не является основной сутью алгоритма).

3.3. *Листинг программы с комментариями*. Копируется весь программный код.

3.4. *Тестирование программы*. Составляется таблица, содержащая следующие поля: номер теста, входные данные, результат выполнения программы, корректность, время выполнения (мс), затраченная память (Мб). Составляется не менее 10-ти тестовых наборов данных согласно условию задачи. Тестовые наборы входных данных студент составляет самостоятельно. В обязательном порядке программа тестируется на граничных наборах входных данных (например, если N варьируется от 0 до 10^9 , то обязательно рассмотреть решение задачи при $N=0$ и при N близком к 10^9). Если написанная программа не позволяет решить задачу при граничных входных данных, все равно включить в тест и в качестве результата написать "Не решено". В столбце "входные данные" данные вносятся вручную, в столбце "результат..." представляется скриншот выполнения программы (если не влезает на одну страницу, делать несколько скриншотов).

Каждая лабораторная работа защищается на занятии преподавателю.

Задачи по теме 3. Элементарные структуры данных

Задача 1. Стек

Ограничение по времени – 2 секунды. Ограничение по памяти – 256 мегабайт.

Реализуйте работу стека. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо "+ N", либо "-". Команда "+ N" означает добавление в стек числа N, по модулю не превышающего 109. Команда "-" означает изъятие элемента из стека. Гарантируется, что не происходит извлечения из пустого стека. Гарантируется, что размер стека в процессе выполнения команд не превысит 106 элементов.

Формат входного файла

В первой строке входного файла содержится M ($1 \leq M \leq 106$) — число команд. Каждая последующая строка исходного файла содержит ровно одну команду.

Формат выходного файла

Выведите числа, которые удаляются из стека с помощью команды "-", по одному в каждой строке. Числа нужно выводить в том порядке, в котором они были извлечены из стека. Гарантируется, что изъятий из пустого стека не производится.

Задача 2. Очередь

Ограничение по времени – 2 секунды. Ограничение по памяти – 256 мегабайт.

Реализуйте работу очереди. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо «+ N», либо «-». Команда «+ N» означает добавление в очередь числа N, по модулю не превышающего 109. Команда «-» означает изъятие элемента из очереди. Гарантируется, что размер очереди в процессе выполнения команд не превысит 106 элементов.

Формат входного файла

В первой строке содержится M ($1 \leq M \leq 106$) — число команд. В последующих строках содержатся команды, по одной в каждой строке.

Формат выходного файла

Выведите числа, которые удаляются из очереди с помощью команды «-», по одному в каждой строке. Числа нужно выводить в том порядке, в котором они были извлечены из очереди. Гарантируется, что извлечения из пустой очереди не производится.

Задача 3. Скобочная последовательность

Последовательность A, состоящую из символов из множества «(», «)», «[» и «]», назовем *правильной скобочной последовательностью*, если выполняется одно из следующих утверждений:

A — пустая последовательность;

первый символ последовательности A — это «(», и в этой последовательности существует такой символ «)», что последовательность можно представить как $A=(B)C$, где B и C — правильные скобочные последовательности;

первый символ последовательности A — это «[», и в этой последовательности существует такой символ «]», что последовательность можно представить как $A=[B]C$, где B и C — правильные скобочные последовательности.

Так, например, последовательности « $()$ » и « $()[]$ » являются правильными скобочными последовательностями, а последовательности « $[]$ » и « $(($ » таковыми не являются.

Входной файл содержит несколько строк, каждая из которых содержит последовательность символов « $($ », « $)$ », « $[$ » и « $]$ ». Для каждой из этих строк выясните, является ли она правильной скобочной последовательностью.

Формат входных данных:

Первая строка входного файла содержит число N ($1 \leq N \leq 500$) - число скобочных последовательностей, которые необходимо проверить. Каждая из следующих N строк содержит скобочную последовательность длиной от 1 до 104 включительно. В каждой из последовательностей присутствуют только скобки указанных выше видов.

Формат выходных данных:

Для каждой строки входного файла выведите в выходной файл «YES», если соответствующая последовательность является правильной скобочной последовательностью, или «NO», если не является.

Задача 4. Очередь с приоритетами

Реализуйте очередь с приоритетами. Ваша очередь должна поддерживать следующие операции: добавить элемент, извлечь минимальный элемент, уменьшить элемент, добавленный во время одной из операций.

Формат входных данных:

В первой строке входного файла содержится число n ($1 \leq n \leq 106$) - число операций с очередью.

Следующие n строк содержат описание операций с очередью, по одному описанию в строке. Операции могут быть следующими:

$A\ x$ — требуется добавить элемент x в очередь.

X — требуется удалить из очереди минимальный элемент и вывести его в выходной файл. Если очередь пуста, в выходной файл требуется вывести звездочку «*».

$D\ x\ y$ — требуется заменить значение элемента, добавленного в очередь операцией A в строке входного файла номер $x+1$, на y . Гарантируется, что в строке $x+1$ действительно находится операция A , что этот элемент не был ранее удален операцией X , и что y меньше, чем предыдущее значение этого элемента.

В очередь помещаются и извлекаются только целые числа, не превышающие по модулю 109.

Формат выходных данных:

Выведите последовательно результат выполнения всех операций X , по одному в каждой строке выходного файла. Если перед очередной операцией X очередь пуста, выведите вместо числа звездочку «*».

Задача 5. Куча ли?

Структуру данных «куча», или, более конкретно, «неубывающая пирамида», можно реализовать на основе массива. Для этого должно выполняться основное свойство

неубывающей пирамиды, которое заключается в том, что для каждого $1 \leq i \leq n$ выполняются условия:

- если $2i \leq n$, то $a[i] \leq a[2i]$;
- если $2i+1 \leq n$, то $a[i] \leq a[2i+1]$.

Дан массив целых чисел. Определите, является ли он неубывающей пирамидой.

Формат входных данных:

Первая строка входного файла содержит целое число n ($1 \leq n \leq 106$). Вторая строка содержит n целых чисел, по модулю не превосходящих $2 \cdot 10^9$.

Формат выходных данных:

Выведите «YES», если массив является неубывающей пирамидой, и «NO» в противном случае.

Задача 6. Очередь с минимумом

Реализуйте работу очереди. В дополнение к стандартным операциям очереди, необходимо также отвечать на запрос о минимальном элементе из тех, которые сейчас находятся в очереди. Для каждой операции запроса минимального элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо «+ N», либо «-», либо «?». Команда «+ N» означает добавление в очередь числа N, по модулю не превышающего 10^9 . Команда «-» означает изъятие элемента из очереди. Команда «?» означает запрос на поиск минимального элемента в очереди.

Формат входных данных:

В первой строке содержится M ($1 \leq M \leq 106$) — число команд. В последующих строках содержатся команды, по одной в каждой строке.

Формат выходных данных:

Для каждой операции поиска минимума в очереди выведите её результат. Результаты должны быть выведены в том порядке, в котором эти операции встречаются во входном файле. Гарантируется, что операций извлечения или поиска минимума для пустой очереди не производится.

Задача 7. Quack

Ограничение по времени – 2 секунды. Ограничение по памяти – 256 мегабайт.

Язык Quack — забавный язык, который фигурирует в одной из задач с [Internet Problem Solving Contest](#). В этой задаче вам требуется написать интерпретатор языка Quack.

Виртуальная машина, на которой исполняется программа на языке Quack, имеет внутри себя очередь, содержащую целые числа по модулю 65536 (то есть, числа от 0 до 65535, соответствующие беззнаковому 16-битному целому типу). Слово get в описании операций означает извлечение из очереди, put — добавление в очередь. Кроме того, у виртуальной машины есть 26 регистров, которые обозначаются буквами от 'a' до 'z'. Изначально все регистры хранят нулевое значение. В языке Quack существуют следующие команды (далее под α и β подразумеваются некие абстрактные временные переменные):

+	Сложение: get α , get β , put $(\alpha + \beta) \bmod 65536$
-	Вычитание: get α , get β , put $(\alpha - \beta) \bmod 65536$
*	Умножение: get α , get β , put $(\alpha \cdot \beta) \bmod 65536$

/	Целочисленное деление: get α , get β , put $\alpha \div \beta$ (будем считать, что $\alpha \div 0 = 0$)
%	Взятие по модулю: get α , get β , put $\alpha \bmod \beta$ (будем считать, что $\alpha \bmod 0 = 0$)
>[register]	Положить в регистр: get α , установить значение [register] в α
<[register]	Взять из регистра: put значение [register]
P	Напечатать: get α , вывести α в стандартный поток вывода и перевести строку
P[register]	Вывести значение регистра [register] в стандартный поток вывода и перевести строку
C	Вывести как символ: get α , вывести символ с ASCII-кодом $\alpha \bmod 256$ в стандартный поток вывода
C[register]	Вывести регистр как символ: вывести символ с ASCII-кодом $\alpha \bmod 256$ (где α — значение регистра [register]) в стандартный поток вывода
:[label]	Метка: эта строка программы имеет метку [label]
J[label]	Переход на строку с меткой [label]
Z[register][label]	Переход если 0: если значение регистра [register] равно нулю, выполнение программы продолжается с метки [label]
E[register1][register2][label]	Переход если равны: если значения регистров [register1] и [register2] равны, исполнение программы продолжается с метки [label]
G[register1][register2][label]	Переход если больше: если значение регистра [register1] больше, чем значение регистра [register2], исполнение программы продолжается с метки [label]
Q	Завершить работу программы. Работа также завершается, если выполнение доходит до конца программы
[number]	Просто число во входном файле — put это число

Формат входного файла

Входной файл содержит синтаксически корректную программу на языке Quack. Известно, что программа завершает работу не более чем за 105 шагов. Программа содержит не менее одной и не более 105 инструкций. **Метки имеют длину от 1 до 10 и состоят из цифр и латинских букв.**

Формат выходного файла

Выведите содержимое стандартного потока вывода виртуальной машины в выходной файл.

Задача 8. Расстояние в дереве

Дано взвешенное дерево. Найти кратчайшее расстояние между заданными вершинами.

Формат входных данных:

Первая строка содержит целое число n — количество вершин в дереве ($1 \leq n \leq 50000$). Вершины нумеруются целыми числами от 0 до $n - 1$. В следующих $n - 1$ строках содержится

по три целых числа u , v , w , которые соответствуют ребру весом w ($0 \leq w \leq 1000$), соединяющему вершины u и v . В следующей строке содержится целое число m — количество запросов ($1 \leq m \leq 75000$). В следующих m строках содержится по два числа — номера вершин, расстояние между которыми необходимо вычислить.

Формат выходных данных:

Для каждого запроса выведите на отдельной строке одно число — искомое расстояние.

Задача 9. Вложенные отрезки

Ограничение времени: 1 секунда. Ограничение памяти: 64 МБ.

На прямой лежат n отрезков. Для каждой пары отрезков известно, что они либо не имеют общих точек, либо все точки одного из них также принадлежат и другому отрезку.

Дано m запросов. Каждый запрос представляет собой точку на прямой. Найдите для каждого запроса отрезок минимальной длины, которому принадлежит эта точка.

Формат входных данных:

В первой строке записано целое число n — количество отрезков ($1 \leq n \leq 10^5$). i -я из следующих n строк содержит целые числа a_i и b_i — координаты концов i -го отрезка ($1 \leq a_i < b_i \leq 10^9$). Отрезки упорядочены по неубыванию a_i , а при $a_i = a_j$ — по убыванию длины. Совпадающих отрезков нет. В следующей строке записано целое число m — количество запросов ($1 \leq m \leq 10^5$). В j -й из следующих m строк записано целое число c_j — координата точки ($1 \leq c_j \leq 10^9$). Запросы упорядочены по неубыванию c_j .

Формат выходных данных:

Для каждого запроса выведите номер искомого отрезка в отдельной строке. Если точка не принадлежит ни одному отрезку, выведите «-1». Отрезки пронумерованы числами от 1 до n в том порядке, в котором они перечислены во входных данных.

Задача 10. Столовая

Ограничение времени: 1 секунда. Ограничение памяти: 64 МБ

Опасно есть в столовой — можно отравиться несвежими продуктами, и вообще — опасно. Причём один человек может впасть в кому от столовской курицы, а другому хоть бы хны. И наоборот. Еду в столовой готовят из M разных продуктов. Всего в меню N продуктов, но не все они есть на раздаче. Допустим, что едят эту еду $K + 1$ студентов и для каждого из них известно, какими продуктами он может отравиться. Первым, естественно, ест самый хитрый — тот, кто пролез без очереди. И он, допустим, не отравился. Как тогда обед подействует на остальных?

Формат входных данных:

В первой строке находится одно число N ($1 \leq N \leq 100$). В следующих N строках названия продуктов — непустые последовательности латинских букв и цифр длиной не более 40 символов. Затем следует число K ($1 \leq K \leq 100$), за которым идёт $K + 1$ блоков, описывающих продукты из меню, опасные для посетителей столовой. i -й такой блок начинается строкой с числом N_i — количеством продуктов, вслед за которым идёт N_i строк с названиями опасных продуктов ($0 \leq N_i \leq N$). Первый блок описывает продукты, опасные для самого хитрого студента, следующие K блоков — для всех остальных. Вход заканчивается строкой, содержащей число M ($0 \leq M \leq N$).

Формат выходных данных:

Выведите K строк — в i -й строке:

YES, если обед будет полностью безвреден для $(i + 1)$ -го студента,
NO, если среди имеющихся продуктов есть вредный для $(i + 1)$ -го студента,
MAYBE, если при таких исходных данных возможна и та, и другая ситуация

Задача 11. Четность

Ограничение времени: 2 секунды. Ограничение памяти: 64 МБ

Вы играете со своим другом в следующую игру. Ваш друг записывает последовательность, состоящую из нулей и единиц. Вы выбираете непрерывную подпоследовательность (например, подпоследовательность от третьей до пятой цифры включительно) и спрашиваете его, чётное или нечётное количество единиц содержит эта подпоследовательность. Ваш друг отвечает, после чего вы можете спросить про другую подпоследовательность, и так далее.

Ваша задача — угадать всю последовательность чисел. Но вы подозреваете, что некоторые из ответов вашего друга могут быть неверными, и хотите уличить его в обмане. Вы решили написать программу, которая получит наборы ваших вопросов вместе с ответами друга и найдет первый ответ, который гарантированно неверен. Это должен быть такой ответ, что существует последовательность, удовлетворяющая ответам на предыдущие вопросы, но никакая последовательность не удовлетворяет этому ответу.

Формат входных данных:

Ввод содержит несколько тестов. Первая строка каждого теста содержит одно число, равное длине последовательности нулей и единиц. Эта длина не превосходит 10^9 . Во второй строке находится одно неотрицательное целое число — количество заданных вопросов и ответов на них. Количество вопросов и ответов не превышает 5 000. Остальные строки содержат вопросы и ответы. Каждая строка содержит один вопрос и ответ на этот вопрос: два целых числа (позиции первой и последней цифр выбранной подпоследовательности) и одно слово — “even” или “odd” — ответ, сообщающий чётность количества единиц в выбранной подпоследовательности, где “even” означает чётное количество единиц, а “odd” означает нечётное количество. Ввод заканчивается строкой, содержащей -1.

Формат выходных данных:

Каждая строка вывода должна содержать одно целое число X . Число X показывает, что существует последовательность нулей и единиц, удовлетворяющая первым X условиям чётности, но не существует последовательности, удовлетворяющей $X + 1$ условию. Если существует последовательность нулей и единиц, удовлетворяющая всем заданным условиям, то число X должно быть равно количеству всех заданных вопросов.

Задача 12. Дерево

Ограничение времени: 1 секунда. Ограничение памяти: 64 МБ.

Рассмотрим дерево, состоящее из n вершин. Назовём *расстоянием* между двумя вершинами минимальное количество рёбер в пути, соединяющем эти вершины. По вершине v_i и расстоянию d_i найдите такую вершину u_i , что расстояние между v_i и u_i равняется d_i .

Формат входных данных:

В первой строке записано количество вершин n ($1 \leq n \leq 20000$) и количество запросов q ($1 \leq q \leq 50000$). Каждая из следующих $n - 1$ строк описывает ребро и содержит номера вершин, соединённых этим ребром. Вершины занумерованы числами от 1 до n . В

следующих q строках заданы запросы. Каждый запрос представляет собой строку, в которой записаны числа v_i ($1 \leq v_i \leq n$) и d_i ($0 \leq d_i \leq n$).

Формат выходных данных:

Выведите q строк. В i -й строке выведите номер вершины u_i — ответ на i -й запрос. Если существует несколько возможных ответов, выведите любой из них. Если искомой вершины не существует, выведите 0.

Задача 13. Постфиксная запись

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как $A B +$. Запись $B C + D * \text{ обозначает привычное нам } (B + C) * D$, а запись $A B C + D * + \text{ означает } A + (B + C) * D$. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Дано выражение в обратной польской записи. Определите его значение.

Формат входных данных:

В первой строке входного файла дано число N ($1 \leq N \leq 10^6$) - число элементов выражения. Во второй строке содержится выражение в постфиксной записи, состоящее из N элементов. В выражении могут содержаться неотрицательные однозначные числа и операции $+$, $-$, $*$. Каждые два соседних элемента выражения разделены ровно одним пробелом.

Формат выходных данных:

Необходимо вывести значение записанного выражения. Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений, по модулю будут меньше, чем 231.

Задача 14. Миллиардеры

Ограничение времени: 3 секунды. Ограничение памяти: 64 МБ

Возможно, вы знаете, что из всех городов мира больше всего миллиардеров живёт в Москве. Но, поскольку работа миллиардера подразумевает частые перемещения по всему свету, в определённые дни какой-то другой город может занимать первую строчку в таком рейтинге. Ваши приятели из ФСБ, ФБР, МИ5 и Шин Бет скинули вам списки перемещений всех миллиардеров за последнее время. Ваш работодатель просит посчитать, сколько дней в течение этого периода каждый из городов мира был первым по общей сумме денег миллиардеров, находящихся в нём.

Формат входных данных:

В первой строке записано число n — количество миллиардеров ($1 \leq n \leq 10000$). Каждая из следующих n строк содержит данные на определённого человека: его имя, название города, где он находился в первый день данного периода, и размер состояния. В следующей строке записаны два числа: m — количество дней, о которых есть данные ($1 \leq m \leq 50000$), k — количество зарегистрированных перемещений миллиардеров ($0 \leq k \leq 50000$). Следующие k строк содержат список перемещений в формате: номер дня (от 1 до $m - 1$), имя человека, название города назначения. Вы можете считать, что миллиардеры путешествуют не чаще одного раза в день, и что они отбывают поздно вечером и прибывают в город назначения рано утром следующего дня. Список упорядочен по возрастанию номера дня. Все имена и названия городов состоят не более чем из 20 латинских букв,

регистр букв имеет значение. Состояния миллиардеров лежат в пределах от 1 до 100 миллиардов.

Формат выходных данных:

В каждой строке должно содержаться название города и, через пробел, количество дней, в течение которых этот город лидировал по общему состоянию миллиардеров, находящихся в нём. Если таких дней не было, пропустите этот город. Города должны быть отсортированы по алфавиту (используйте обычный порядок символов: ABC...Zabc...z).

Задача 15. Провода

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

На складе есть провода различной целочисленной длины. Их можно разрезать на части. Необходимо получить K кусочков одинаковой целочисленной и как можно большей длины. Найти максимальную длину M , при которой можно получить по меньшей мере K кусочков этой длины. Все оставшиеся на складе куски проводов длиной меньше M в подсчете не участвуют.

Формат входных данных:

В первой строке – количество проводов на складе N и необходимое количество кусочков K . В следующих N строках – длины проводов.

$$1 \leq N \leq 100000$$

$$1 \leq K \leq 100000$$

Формат выходных данных:

M – максимальная длина, на которую можно разрезать все провода так, чтобы получилось не менее K кусочков.

Примеры:

Стандартный ввод	Стандартный вывод
5 7 15 12 5 13 6	6

Задача 16. Брокеры

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

В стране Бурляндии фирма «Котлетный рай» имеет много отделений, работающих сравнительно автономно. После неких экономических преобразований такая форма функционирования оказалась невыгодной и фирма решила сливать капиталы отделений, образуя укрупненные департаменты, отвечающие за несколько отделений сразу. Цель фирмы – слить все отделения в один громадный департамент, владеющий всеми капиталами. Проблема заключается в том, что по законам Бурляндии операция слияния капиталов должна проводиться государственной брокерской службой, которая не может производить более одной операции слияния в одной фирме одновременно. Вторая проблема состоит в том, что брокерская служба берет за свои услуги один процент всех средств,

получившихся в результате слияния двух подразделений. Важно спланировать порядок операций слияния таким образом, чтобы фирма потратила на слияние как можно меньшую сумму.

Формат входных данных:

На вход программы подается число отделение $2 \leq N \leq 1000000$, за которым следует N капиталов отделений $1 \leq C_i \leq 1000000$.

Формат выходных данных:

T – минимальная сумма из возможных, которую должна заплатить брокерам фирма «Котлетный рай», с двумя знаками после запятой.

Примеры:

Стандартный ввод	Стандартный вывод
5 1 2 3 4 5	0.33
10 2 10 100 30 7 4 15 2 15 80	6.52

Задача 17. Пересечение множеств

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Задано $2 \leq N \leq 1000$ множеств из $3 \leq M \leq$ элементов, значения которых могут находиться в диапазоне от -2000000000 до 2000000000. Значения каждого множества задаются в произвольном порядке. Гарантируется, что для одного множества все задаваемые значения различные.

Требуется найти наибольший размер множества, получаемого при пересечении какой-либо из пар из заданных множеств.

Формат входных данных:

N M

$A_1[1] A_1[2] \dots A_1[M]$

$A_2[1] A_2[2] \dots A_2[M]$

...

$A_N[1] A_N[2] \dots A_N[M]$

Формат выходных данных:

Одно целое число.

Примеры:

Стандартный ввод	Стандартный вывод
3 4 9 7 1 8 5 7 6 3 5 9 8 6	2
4 5 -2 6 8 4 -1 5 3 10 -5 -1 7 8 -5 -1 -2 -1 8 4 9 0	3

Задача 18. Составные строки

Ограничение по времени: 0.5 секунд

Ограничение по памяти: 16 мегабайт

В первой строке входного файла содержится число $4 \leq N \leq 1000$, последующие N строк состоят только из заглавных букв латинского алфавита и образуют множество, то есть, равных элементов среди них нет. Длина строки не превышает 1000 букв. Некоторые из этих строк можно составить, прописав друг за другом каких-то два элемента множества, возможно, совпадающие. То есть, если исходное множество содержит элементы A, B, AB, AA, C, ABC, то элемент AB можно составить из элементов A и B, а строку AA – из элементов A и A.

Ваша задача – вывести все элементы множества, которые можно составить из пары элементов множества по одному на строку в лексикографически возрастающем порядке.

Примеры:

Стандартный ввод	Стандартный вывод
5 A AB B AA ABC	AA AB
10 ABC DEFG AB ABCAB DEFGA FG ABFG ABCAFG FGFG ABABC	ABCAB ABFG FGFG

Задача 19. Длинное деление

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

На вход подается три строки. Первая содержит представление длинного десятичного числа (первый операнд), вторая – всегда строка /, третья – десятичное представление второго операнда.

Длина первой и третьей строки ограничены 100000 символами. Вторая строка содержит ровно один символ /.

Требуется исполнить операцию и вывести результат в десятичном представлении.

Для отрицательных делимого и делителя действуют правила математики.

$$27 / 10 = 2$$

$$-27 / 10 = -3$$

$$27 / -10 = -3$$

$$-27 - -10 = 2$$

Формат входных данных:

999

/

9

Формат выходных данных:

111

Пример

стандартный ввод	стандартный вывод
11876187623876123293872987 / 1987987198711	5973975904662

Задача 20. Ксерокопии

Ограничение по времени: 0,2 секунды

Ограничение по памяти: 16 мегабайт

Секретарша Людоча сегодня опоздала на работу и ей срочно нужно успеть к обеду сделать N копий одного документа. В ее распоряжении имеются два ксерокса, один из которых копирует лист за x секунд, а другой – за y секунд. (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ей выяснить, какое минимальное время для этого потребуется.

Формат входных данных:

Во входном файле INPUT.TXT записаны три натуральных числа N, x и y, разделенные пробелом ($1 \leq N \leq 2 \cdot 10^8$, $1 \leq x, y \leq 10$).

Формат выходных данных:

В выходной файл OUTPUT.TXT выведите одно число – минимальное время в секундах, необходимое для получения N копий.

Пример

INPUT.TXT	OUTPUT.TXT
4 1 1	3
5 1 2	4

Задача 21. Роман в томах

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

В романе N глав. В i -той главе a_i страниц. Требуется издать роман в K томах так, чтобы объем самого «толстого» тома был минимален. В каждом томе главы располагаются по порядку своих номеров.

Требуется написать программу, которая найдет количество страниц в самом «толстом» томе.

Формат входных данных:

Входной текстовый файл INPUT.TXT содержит в первой строке число N ($1 \leq N \leq 100$). Во второй строке через пробел записаны N чисел – количество страниц в каждой главе. Количество страниц в романе не превышает 32767. В третьей строке записано число K ($1 \leq K \leq N$).

Формат выходных данных:

Выходной файл OUTPUT.TXT должен содержать количество страниц в самом «толстом» томе.

Пример

INPUT.TXT	OUTPUT.TXT
3 1 2 1 2	3
4 1 2 1 1 3	2