

Задача 1. Маски имён файлов

Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов. Также в маске может встречаться символ “*”, обозначающий любую последовательность любых символов, возможно пустую. Например, если задана маска “a*po*”, то такой маске удовлетворяют файлы с именами “airport”, “appoint”, “apologize”, но не удовлетворяют, например, файлы с именами “approve”; или “weapon”.

Вам нужно придумать маску имени файла, которой удовлетворяли бы следующие имена файлов:
python_program

и

math_theorem

но при этом **НЕ** удовлетворяли бы файлы с именами

nuclear_synthesis

и

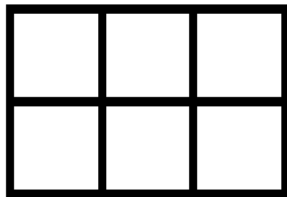
biological_system

Вам нужно придумать как можно более короткую маску (содержащую как можно меньше символов). Чем короче будет придуманная вами маска, тем больше баллов вы получите.

Ответом на эту задачу является строка, которая может содержать строчные буквы английского алфавита, символ “_” (подчёркивание) и символ “*” (звёздочка).

Задача 2. Городские кварталы

План города представляет собой прямоугольник размером $n \times m$ километров, разбитых на кварталы — квадраты со стороной 1 километр. Между кварталами, а также по границе города проложены дороги. Например, если $n = 2$, $m = 3$, то план дорог следующий.



Определите суммарную длину всех дорог в городе в километрах. Единицу измерения указывать не нужно.

Ответом к этой задаче является некоторое выражение, которое может содержать целые числа, переменные n и m , операции сложения (обозначается +), вычитания (обозначается −), умножения (обозначается *) и круглые скобки для изменения порядка действий. Запись вида $2n$ для обозначения произведения числа 2 и переменной n неверная, нужно писать $2 * n$.

Пример правильного (по форме записи) выражения:

$$n + (m - 1) * 2$$

Задача 3. Пятибуквенные последовательности

Составим все возможные последовательности из пяти букв, используя только символы L, R, V, и выпишем их в алфавитном порядке. Вот начало этого списка:

1. LLLLL
2. LLLLR
3. LLLLV
4. LLLRL

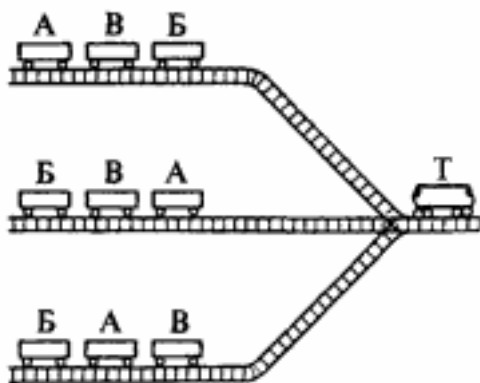
Определите последовательности, которые будут идти в этом списке под номерами 8, 81, 98, 110, 179.

Возможно, вам будет проще ответить на третий и четвертый вопрос если вы будете знать, что на 100-м месте в этом списке стоит строка RLVLL.

В ответе нужно записать пять строк, состоящих из латинских букв. Ответ на каждое задание нужно писать в отдельной строке (в первой строке — слово, стоящее в списке 8-м, во второй строке — слово, стоящее 81-м, в третьей строке — 98-м, в четвертой строке — 110-м, в пятой строке — 179-м). Порядок записи строк в ответе менять нельзя. В ответе должно быть ровно пять строк. Если вы не можете найти какой-то из ответов, вместо него напишите любую строчку из данных пяти букв.

Задача 4. Станция

На трёх путях сортировочной станции стоят вагоны с абрикосами (А), бананами (Б) и вишней (В) так, как это показано на рисунке.



Машинист тепловоза (Т) может за один раз прицепить к тепловозу один или несколько вагонов с какого-нибудь пути, передвинуть их на правый путь, после чего передвинуть их на любой левый путь. Например, если тепловоз заберет 2 вагона с пути номер 3 на путь номер 1, то после такого действия распределение вагонов по путям будет таким.

1. А В Б А В
2. Б В А
3. Б

Соответствующую команду перемещения будем записывать так.

2 3 1

Что означает, что необходимо передвинуть 2 вагона с пути номер 3 на путь номер 1. Первое число обозначает количество перемещаемых вагонов, второе число — номер пути, с которого нужно забрать вагоны, третье число — номер пути, на который нужно передвинуть вагоны.

Разработайте алгоритм действий машиниста, необходимых для того, чтобы сформировать на каждом из путей составы с одинаковыми фруктами (не важно, на каком именно пути). Алгоритм оформите в виде последовательности команд, записанных в отдельных строках. Каждая команда имеет указанный выше вид: сначала записано число передвигаемых вагонов, потом номер пути с которого передвигаются вагоны, затем номер пути, на который передвигаются вагоны.

Например, следующая запись:

2 3 1
1 2 3

означает “передвинуть 2 вагона с пути 3 на путь 1, затем передвинуть 1 вагон с пути 2 на путь 3”.

Чем меньше команд будет в вашем алгоритме, тем большее количество баллов вы получите.

Задача 5. Летоисчисление

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	512 мегабайт

В григорианском календаре года нумеруются числами 1, 2, 3 и т.д., это года “нашей эры”. Предшествующие года называются “первый год до нашей эры”, “второй год до нашей эры” и т.д.

Будем обозначать года нашей эры положительными числами, а года до нашей эры — отрицательными. При этом года с номером 0 не существует, то есть нумерация лет выглядит так: ..., −3, −2, −1, 1, 2, 3, ...

В летописях написано, что какое-то событие произошло в году номер A , а другое событие произошло спустя n лет после первого события (или за n лет до первого события). Определите, в каком году произошло второе событие.

Формат входных данных

Первая строка входных данных содержит число A — год, в котором произошло первое событие. Вторая строка содержит число n . Если $n > 0$, то второе событие произошло через n лет после первого события, а если $n < 0$, то второе событие произошло за $|n|$ лет до первого события. Оба числа могут быть как положительными, так и отрицательными.

Формат выходных данных

Программа должна вывести одно целое число — номер года, в который произошло второе событие.

Система оценивания

Решение, правильно работающее только для случаев, когда все входные числа по модулю не превосходят 100, будет оцениваться в 6 баллов.

В 10 баллов будет оцениваться решение, правильно работающее, когда все входные числа по модулю не превосходят 10^9 .

Примеры

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

Пояснения к примерам

Пояснение к первому примеру. Первое событие произошло в 5 году, второе событие произошло за 3 года до первого, это был 2 год.

Пояснение ко второму примеру. Первое событие произошло в 3 году до н.э., второе событие произошло через 1 год, это 2 год до н.э. Ответ: −2.

Пояснение к третьему примеру. Первое событие произошло в 3 году до н.э., второе событие произошло через 4 года. Отсчитываем 4 года: 2 год до н.э., 1 год до н.э., 1 год н.э., 2 год н.э. Ответ: 2.

Задача 6. Прожектора

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	512 мегабайт

На дискотеке в ряд стоят три прожектора, которые поочерёдно светят в следующем порядке: левый, средний, правый, средний, левый, средний, правый, средний и т.д. (слева направо, затем налево, опять направо, ...). Каждый прожектор горит в течение одной секунды.

Известно, что лампа левого прожектора имеет ресурс A секунд горения, среднего — B секунд, правого — C секунд. Определите, сколько времени сможет продолжаться этот процесс горения прожекторов.

Формат входных данных

Программа получает на вход три целых неотрицательных числа A, B, C — время горения левого, среднего, правого прожектора.

Формат выходных данных

Программа должна вывести одно целое число.

Система оценивания

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 10, будет оцениваться в 4 балла.

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 10^4 , будет оцениваться в 7 баллов.

В 10 баллов будет оцениваться решение, правильно работающее, когда сумма всех входных чисел по модулю не превосходит 2×10^9 .

Пример

стандартный ввод	стандартный вывод
3 3 3	7

Пояснение к примеру

Прожектора горят в следующем порядке: левый, средний, правый, средний, левый, средний, правый. После этого должен загореться средний прожектор, но он уже выработал ресурс и загореться не сможет. Поэтому процесс обрывается после 7 с.

Задача 7. Ремонт забора

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Забор состоит из N одинаковых вертикальных досок. Некоторые из досок сгнили и нуждаются в замене, для каждой доски известно, нужно ли её заменить. Для ремонта забора можно использовать продающиеся в магазине щиты, которые бывают L разных видов: шириной в 1 доску, в 2 доски, ..., в L досок. Щит нельзя разрезать на части, то есть одним щитом можно заменить не более любых L подряд идущих досок. При этом можно менять не только сгнившие доски, но и хорошие.

Оказалось, что все щиты стоят одинаково, независимо от размера щита. Определите, какое наименьшее число щитов необходимо приобрести, чтобы починить весь забор.

Формат входных данных

Первая строка входных данных содержит целое число L ($L > 0$) — максимальный размер щита. Во второй строке входных данных записано целое число N ($N > 0$) — количество досок в заборе. Следующие N строк содержат по одному числу, равному 0 или 1. Число 1 обозначает, что соответствующая доска в заборе нуждается в замене, число 0 — что доска может быть сохранена.

Формат выходных данных

Программа должна вывести одно целое число — минимальное число щитов, которое необходимо приобрести для ремонта всего забора.

Система оценивания

Решение, правильно работающее только для случаев, когда числа L и N не превосходят 1000, будет оцениваться в 6 баллов.

В 10 баллов будет оцениваться решение, правильно работающее, когда числа L и N не превосходят 10^5 .

Пример

стандартный ввод	стандартный вывод
3	3
8	
0	
0	
1	
0	
1	
0	
1	
0	

Пояснение к примеру

Максимальная ширина одного щита равна 3. Забор состоит из 8 досок, нужно заменить доски с номерами 3, 5 и 7. Для этого достаточно двух щитов, например, одним щитом меняем доски с номерами 3, 4, 5, а другим щитом меняем доску с номером 7.