

Задача G. Послезавтра

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

По заданной дате требуется определить, какое число будет послезавтра.

Формат входного файла

Дано число, месяц и год (год — число в промежутке от 1 до 10000).

Формат выходного файла

Требуется вывести, какое число будет послезавтра, в формате входных данных.

Примеры

стандартный ввод	стандартный вывод
1 8 2009	3 8 2009
30 12 2009	1 1 2010
28 2 2008	1 3 2008

Задача H. Заменить цифру

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

Дано число N , а также цифры A и B . Заменить в этом числе каждое вхождение цифры A на цифру B .

Формат входного файла

Программе даны числа N, A, B ($1 \leq N \leq 10^9$).

Формат выходного файла

Вывести полученное число.

Примеры

стандартный ввод	стандартный вывод
123118 1 5	523558

Задача I. Контроперация

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

Хакер Василий получил доступ к классному журналу и хочет заменить все свои минимальные оценки на максимальные. Напишите программу, которая заменяет оценки Василия, но наоборот.

Формат входного файла

Дано количество оценок Василия (не больше 100), затем сами оценки.

Формат выходного файла

Требуется вывести исправленные оценки в том же порядке.

Примеры

стандартный ввод	стандартный вывод
5 1 3 3 3 4	1 3 3 3 1
8 5 4 2 2 4 2 2 5	8 2 4 2 2 4 2 2 2

Задача J. Архивация

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Алгоритм сжатия RLE устроен по следующему принципу. Файл рассматривается как последовательность бит. Непрерывные цепочки нулей и единиц заменяются на число — длину цепочки, максимальная длина которой K ограничена (очень длинные цепочки разбиваются на несколько и кодируются несколькими числами, между которыми вставляются нули). Так же для однозначности первой цепочкой считается всегда цепочка нулей (если последовательность начинается с 1, то первым числом при сжатии пишется 0).

Формат входного файла

На первой строке дано число K ($1 \leq K \leq 1000$). На второй строке дана последовательность нулей и единиц, в конце которой находится число 2. Длина последовательности — натуральное число, не превышающее 10^6 .

Формат выходного файла

Сжатая последовательность (см. примеры).

Примеры

стандартный ввод	стандартный вывод
10 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 2	2 5 3 7
3 0 0 0 0 0 0 0 0 0 0 0 0 0 2	3 0 3 0 3 0 3 0 1
5 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 2	0 5 0 4 5 0 1

Задача K. Билеты на метро - 1

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Билет на одну поездку в метро стоит 15 рубля, билет на 10 поездок стоит 125 рублей, билет на 60 поездок стоит 440 рублей. Пассажир планирует совершить n поездок. Определите, сколько билетов каждого вида он должен приобрести, чтобы суммарное количество оплаченных поездок было не меньше n , а общая стоимость приобретенных билетов — минимальна. Программа получает на вход одно число n и должна вывести три целых числа, равных необходимому количеству билетов на 1, на 10, на 60 поездок.

Формат входного файла

На первой строке дано число n ($1 \leq n \leq 1000$).

Формат выходного файла

Программа должна вывести три целых числа, равных необходимому количеству билетов на 1, на 10, на 60 поездок.

Примеры

стандартный ввод	стандартный вывод
12	2 1 0

Задача L. Билеты на метро - 1

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	64 мегабайта

Билет на одну поездку в метро стоит 15 рубля, билет на 10 поездок стоит 125 рублей, билет на 60 поездок стоит 440 рублей. Пассажир планирует совершить p поездок. Определите, сколько билетов каждого вида он должен приобрести, чтобы суммарное количество оплаченных поездок было не меньше p , а общая стоимость приобретенных билетов — минимальна. Программа получает на вход одно число p и должна вывести три целых числа, равных необходимому количеству билетов на 1, на 10, на 60 поездок.

Билет на одну поездку в метро стоит: 1 поездка — 15 рублей, 5 поездок — 70 рублей, 10 поездок — 125 рублей, 20 поездок — 230 рублей, 60 поездок — 440 рублей. Пассажир планирует совершить p поездок. Определите, сколько билетов каждого вида он должен приобрести, чтобы суммарное количество оплаченных поездок было не меньше p , а общая стоимость приобретенных билетов — минимальна.

Программа получает на вход одно число p и должна вывести пять целых чисел: количество билетов на 1, 5, 10, 20, 60 поездок, которое необходимо приобрести. Если для какого-то данного p существует несколько способов приобретения билетов одинаковой суммарной стоимости, необходимо вывести ту комбинацию билетов, которая дает большее число поездок.

Формат входного файла

На первой строке дано число n ($1 \leq n \leq 1000$).

Формат выходного файла

Программа должна вывести три целых числа, равных необходимому количеству билетов на 1, 5, 10, 20, 60 поездок, которое необходимо приобрести.

Примеры

стандартный ввод	стандартный вывод
14	4 0 1 0 0

Задача M. Таймер

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	64 мегабайта

Таймер — это часы, которые умеют подавать звуковой сигнал по прошествии некоторого периода времени. Напишите программу, которая определяет, когда должен быть подан звуковой сигнал.

Формат входного файла

В первой строке входного файла записано текущее время в формате ЧЧ:ММ:СС (с ведущими нулями). При этом оно удовлетворяет ограничениям: ЧЧ - от 00 до 23,

ММ и СС - от 00 до 60. Во второй строке записан интервал времени, который должен быть измерен. Интервал записывается в формате Ч:М:С (где Ч, М и С - от 0 до 109, без ведущих нулей). Дополнительно если Ч=0 (или Ч=0 и М=0), то они могут быть опущены. Например, 100:60 на самом деле означает 100 минут 60 секунд, что то же самое, что 101:0 или 1:41:0. А 42 обозначает 42 секунды. 100:100:100 - 100 часов, 100 минут, 100 секунд, что то же самое, что 101:41:40.

Формат выходного файла

В выходной файл выведите в формате ЧЧ:ММ:СС время, во сколько прозвучит звуковой сигнал. При этом если сигнал прозвучит не в текущие сутки, то дальше должна следовать запись $+<\text{кол-во}> \text{ days}$. Например, если сигнал прозвучит на следующий день - то $+1 \text{ days}$.

Примеры

стандартный ввод	стандартный вывод
01:01:01 48:0:0	01:01:01+2 days
01:01:01 58:119	02:01:00
23:59:59 1	00:00:00+1 days

Задача N. Клад

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	64 мегабайта

Найти закопанный пиратами клад просто: всё, что для этого нужно - это карта. Как известно, пираты обычно рисуют карты от руки и описывают алгоритм нахождения клада так: «Встаньте около одинокой пальмы. Пройдите тридцать шагов в сторону леса, потом семнадцать шагов в сторону озера, ..., наконец десять шагов в сторону большого булыжника. Клад находится под ним». Большая часть таких указаний просто сводится к прохождению какого-то количества шагов в одном из восьми направлений (1 - север, 2 - северо-восток, 3 - восток, 4 - юго-восток, 5 - юг, 6 - юго-запад, 7 - запад, 8 - северо-запад). Длина шага в любом направлении равна 1.

Путешествие по такому пути обычно является прекрасным способом посмотреть окрестности, однако в наше время постоянной спешки ни у кого нет времени на это. Поэтому кладоискатели хотят идти напрямую в точку, где зарыт клад. Например, вместо того, чтобы проходить три шага на север, один шаг на восток, один шаг на север, три шага на восток, два шага на юг и один шаг на запад, можно пройти напрямую, использовав около 3.6 шага.

Вам необходимо написать программу, которая по указаниям пиратов определяет точку, где зарыт клад.

Формат входного файла

Первая строка входного файла содержит число N - число указаний ($1 \leq N \leq 40$). Последующие N строк содержат сами указания - номер направления (целое число от 1 до 8) и количество шагов (целое число от 1 до 1000). Числа разделены пробелами.

Формат выходного файла

В выходной файл выведите координаты X и Y точки (два вещественных числа, разделённые пробелом), где закрыт клад, считая, что ось Ox направлена на восток, а ось Oy – на север. В начале кладоискатель должен стоять в начале координат. Координаты необходимо вывести с погрешностью не более 0.001.

Примеры

стандартный ввод	стандартный вывод
6	3.000 2.000
1 3	
3 1	
1 1	
3 3	
5 2	
7 1	
1	-7.071 7.071
8 10	

Задача O. Шифр Юлия

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

Юлий Цезарь использовал свой способ шифрования текста. Каждая буква заменялась на следующую по алфавиту через K позиций по кругу. Необходимо по заданной шифровке определить исходный текст.

Формат входного файла

В первой строке дана шифровка, состоящая из главных латинских букв. Во второй строке число K ($1 \leq K \leq 10$).

Формат выходного файла

Требуется вывести результат расшифровки.

Примеры

стандартный ввод	стандартный вывод
XPSE	WORD
1	
ZABC	WXYZ
3	

Задача P. Лишние пробелы

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

Дана строка. Напишите программу, которая удалит из этой строки все лишние пробелы. Пробел будем считать лишним, если:

- он находится в самом начале строки, до самого первого слова;
- он находится в конце строки, после самого последнего слова;

- несколько пробелов расположены между двумя словами (проще говоря, если слова разделены более чем одним пробелом, тогда все пробелы кроме одного – лишние)

Формат входного файла

Дана строка S ($0 \leq |S| \leq 255$). Строка содержит только строчные латинские буквы и пробелы.

Формат выходного файла

Требуется вывести строку без лишних пробелов.

Примеры

стандартный ввод	стандартный вывод
Alexandr Sergeevich Pushkin	Alexandr Sergeevich Pushkin

Задача Q. Количество дней от начала эры

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

Требуется посчитать количество дней от начала эры до данного дня включительно. Началом эры считается первое января первого года.

Формат входного файла

В единственной строке входного файла находится дата в формате ДДММГГГГ.

Формат выходного файла

Выведите искомое количество дней.

Примеры

стандартный ввод	стандартный вывод
02010001	2
01010002	366

Задача R. Красивые даты

Имя входного файла: c.in
Имя выходного файла: c.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вова считает, что дата, записанная в формате ДДММГГГГ – красивая, если при прочтении её с конца также получается дата, которая существует в календаре. Например, дата первое сентября 2010 года (01092010) является красивой, так как при ее прочтении с конца получается первое февраля 9010 года (01029010), а дата третье апреля 1991 года (03041991) – нет, так как в 4030 году нет 91-го месяца. Его заинтересовал вопрос, сколько дат с дня его рождения по день, когда он пошел в первый класс, обладают таким свойством. Помогите ему.

Формат входного файла

В первой строке входного файла записана дата рождения Вовы в формате ДДММГГГГ, во второй строке записана дата, когда он пошел в первый класс, в таком же формате. Обе являются корректными датами с первого января первого года (записывается как 01010001) по 31 декабря 9999 года (записывается как 31129999).

Формат выходного файла

В выходной файл выведите количество красивых дат.

Примеры

c.in	c.out
03041991 01012001	1
01010001 31129999	133440