1

12-часовой формат времени

2

Числа через запятую

3

Значение дроби

4

IP-адрес

5

Диаметр множества

6

Личные дела

Задача 1.

Вводная

**12-часовой формат времени**

Во многих англоговорящих странах время записывается в 12-часовом формате с указанием одного из двух 12-часовых интервалов: до полудня (a.m.) или после полудня (p.m.). Формально запись времени в 12-часовом формате выглядит, как строка “h:mm x.m.”, где h — число часов от 1 до 12 (однозначное или двухзначное), mm — число минут от 00 до 59, дополненное нулями до двух разрядов, x — одна из двух букв “a” или “p”. Подробней примеры записи времени и их перевод в 24-часовой формат можно найти [в википедии](https://ru.wikipedia.org/wiki/12-%D1%87%D0%B0%D1%81%D0%BE%D0%B2%D0%BE%D0%B9_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8).

Дана строка с записью некоторого времени в 12-часовом формате. Выведите его в 24-часовом формате “hh:mm” (hh — количество часов от 00 до 23, дополненное нулями до двух разрядов, mm — количество минут от 00 до 59, дополненное нулями до двух разрядов).

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 1:00 a.m. | 01:00 |
| 12:34 p.m. | 12:34 |

Начало формы

Задача решена верно

OK. Your score is = 100, 74/74 tests passed

Решение задачи

Сначала нужно считать данные — для этого считываем число, двоеточие, число, двоеточие, символ. Этого достаточно, чтобы получить ответ.

Результат удобней выводить при помощи функции printf со следующим форматом — число выводится в поле шириной 2 символа с заполнением нулями.

Конец формы

Задача 2.

Лёгкая

**Числа через запятую**

Во входной строке записаны целые числа через запятую. Найдите их сумму.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход строку, содержащую цифры, знаки +/- и запятые. Необходимо разбить эту строку по запятым на числа и вычислить их сумму.

Длина строки не превосходит 1000 символов. Строка содержит хотя бы одно число, начинается и заканчивается не с запятой, никакие две запятые не идут подряд. Между двумя соседними запятыми записано корректное целое число, не превосходящее по модулю 106, которое может начинаться со знака "+" или "-".

ВЫХОДНЫЕ ДАННЫЕ

Программа должна вывести одно число — сумму данных чисел.

ПРИМЕР

|  |  |
| --- | --- |
| ввод | вывод |
| 12,-5,+8 | 15 |

Начало формы

Задача решена верно

OK. Your score is = 100, 31/31 tests passed

Решение задачи

Можно считать первое число при помощи scanf("%d"), а последующие числа читать в цикле при помощи scanf(",%d").

Для проверки окончания ввода можно использовать значение, возвращаемое функцией scanf — пока функция возвращает число 1, считывание успешно.

Конец формы

Задача 3.

Средняя

**Значение дроби**

Программа получает на вход строку, в которой записано целое число или дробь. Необходимо вывести значение этого числа.

Чтобы послать программе сообщение об окончании потока стандартного ввода необходимо нажать Ctrl+Z в системе Windows или Ctrl+D в системе Linux.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход строку, которая содержит одно из трех возможных выражений:

a

a/b

a b/c

где a, b, c — целые натуральные числа, не превосходящие 1000. Определите значение данного числа (целого или дроби) и выведите его, как действительное число.

ВЫХОДНЫЕ ДАННЫЕ

Программа должна вывести одно действительное число. Разрешается выводить нули в конце числа.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 7 | 7.000000 |
| 1/3 | 0.333333 |
| 1 1/2 | 1.500000 |

Начало формы

Задача решена верно

OK. Your score is = 100, 25/25 tests passed

Решение задачи

Первое число можно считать при помощи строки scanf("%d"). Но после этого числа будет следовать либо пробел, либо "/" (а может и вообще ничего). Поэтому после числа нужно считать один символ. Затем — второе число, опять символ и третье число.

Правильная форматная строка для чтения числа выглядит так: "%d%c%d%c%d"

Для того, чтобы определить, сколько значений было считано, будем использовать значение, которое возвращает функция scanf.

Конец формы

Задача 4.

Олимпиадная

**IP-адрес**

В сети интернет каждому компьютеру присваивается четырехбайтовый код, который принято записывать в виде четырех чисел, каждое из которых может принимать значения от 0 до 255, разделенных точками. Вот примеры правильных IP-адресов:

127.0.0.0

192.168.0.1

255.0.255.255

По данной строке определите, является ли она записью корректного IP-адреса.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход строку из произвольных непробельных символов длиной не более 100.

ВЫХОДНЫЕ ДАННЫЕ

Если эта строка является корректной записью IP-адреса, выведите YES, иначе выведите NO.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 127.0.0.1 | YES |

Начало формы

Задача решена верно

OK. Your score is = 100, 32/32 tests passed

Решение задачи

Нужно считать следующую форматную строку – "%d.%d.%d.%d %с". Обратите внимание на пробел перед "%c" — нужно пропустить все пробельные символы и считать непробельный символ (т.к. иначе конец строки тоже будет считаться пробельным символом). Функция должна вернуть 4 — считано 4 числа, а попытка считывания следующего символа была неудачной.

Затем проверим, что все считанные числа принадлежат отрезку [0, 255].

Конец формы

Задача 5.

Лёгкая

**Диаметр множества**

Вам дано множество точек. Выведите диаметр этого множества.

Диаметр - максимальное расстояние между парой точек из этого множества.

Определите структуру данных Point для точки, в ней сделайте конструктор точки, функцию разности точек для получения вектора и функцию нахождения длины вектора.

ВХОДНЫЕ ДАННЫЕ

В первой строке вводится число N - количество точек. (2≤N≤100).

На следующих N строках вводятся координаты точек. В очередной строке вводятся два числа - координаты точки. Координаты не превышают 100 по модулю.

ВЫХОДНЫЕ ДАННЫЕ

Выведите единственное число - диаметр множества. Точность вывода - 15 знаков.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 2  1 2  2 3 | 1.4142135623731 |

Задача 6.

Средняя

**Личные дела**

Однажды неловкая секретарша перепутала личные дела учащихся.Теперь их снова необходимо упорядочить сначала по классам, а внутри класса по фамилиям.

Определите структуру данных для дела, определите конструктор и оператор сравнения.

ВХОДНЫЕ ДАННЫЕ

В первой строке дано число N(1≤N≤1000) - количество личных дел. Далее для каждого из N учащихся даны следующие данные (каждое в своей строке): фамилия и имя, класс, дата рождения. Фамилия и имя - строки не более чем из 20 символов, класс - строка, состоящая из числа (от 1 до 11) и латинской буквы (от "A" до "Z"), дата рождения - дата в формате "ДД.ММ.ГГ". Гарантируется, что внутри одного класса нет однофамильцев.

ВЫХОДНЫЕ ДАННЫЕ

В выходной файл требуется вывести N строк, в каждой из которых записаны данные по одному учащемуся. Строки должны быть упорядочены сначала по классам, а затем по фамилиям.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3  Sidorov  Sidor  9A  20.07.93  Petrov  Petr  9B  23.10.92  Ivanov  Ivan  9A  10.04.93 | 9A Ivanov Ivan 10.04.93  9A Sidorov Sidor 20.07.93  9B Petrov Petr 23.10.92 |

Начало формы

Задача решена верно

OK. Your score is = 100, 30/30 tests passed

Решение задачи

Честно определите структуру личного дела и определите оператор сравнения. Далее просто запустите встроенную функцию sort.

Конец формы