|  |  |  |
| --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ  И НАУКИ КРАСНОДАРСКОГО КРАЯ  Государственное бюджетное образовательное учреждение  дополнительного образования детей  «ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ДЛЯ ДЕТЕЙ»  350000 г. Краснодар,  ул. Красная, 76  тел. 259-84-01  E-mail: cdodd@mail.ru |  | **Всероссийская олимпиада школьников**  **по информатике**  **2014-2015 учебный год**  **Муниципальный этап**  **9-11 классы, задания**  **Председатель предметно-методической комиссии: Гаркуша О.В., к. ф.-м. н., доцент** |

**Задача 1. (20 баллов)**

|  |  |
| --- | --- |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 Мбайт |

**Поезд.** Известно, сколько времени поезд тратит на проезд между любыми двумя соседними станциями своего маршрута. Известно время отправления из начальной станции. Напишите программу, которая вычислит время отправления электропоезда с каждой станции его маршрута (для последней станции это будет время прибытия — временем стоянки поезда на станциях мы пренебрежем).

**Формат входных данных**

В первой строке задано время отправления поезда с начальной станции. Время задается в следующем формате: сначала идут две цифры, задающие часы (от 00 до 23), далее идет двоеточие, затем идут две цифры, задающие минуты (от 00 до 59). Пробелы внутри строки, задающей время, не допускаются.

Во второй строке входного файла записано натуральное число *N* (2≤*N*≤1000) — количество станций в маршруте электропоезда (включая начальную и конечную станции). В третьей строке записано *N*–1 число — первое из этих чисел задает время следования в минутах от начальной станции до второй станции, второе — время от второй станции до третьей и т.д. Каждое из этих чисел натуральное и не превышает 1000.

**Формат выходных данных**

В выходной файл для каждой станции выведите время прохождения поезда через эту станцию. Время должно быть выведено в том же формате, в каком оно задается во входных данных.

**Пример входных и выходных файлов**

|  |  |
| --- | --- |
| Task1.in | Task1.out |
| 07:00  4  10 5 3 | 07:00  07:10  07:15  07:18 |
| 22:58  5  2 60 43 20 | 22:58  23:00  00:00  00:43  01:03 |

**Задача 2. (30 баллов)**

|  |  |
| --- | --- |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 Мбайт |

**Сумма цифр.** Напишите программу, которая вычисляет сумму первых К цифр 100‑значного числа.

**Формат входных данных**

В первой строке содержится число K (1 ≤ K ≤ 100) — количество цифр.

Во второй строке записано 100‑значное число.

**Формат выходных данных**

Одно число — сумму первых К цифр данного числа.

**Пример входных и выходных файлов**

| Task2.in | Task2.out |
| --- | --- |
| 3  0432967930560642502060973925193896474782636108449939354418921190729056087768919950747686217775346048 | 7 |
| 99  9661760740844565862709784606162562632269542749599316636989050353108923932549215219177092447920403779 | 460 |

**Задача 3. (50 баллов)**

|  |  |
| --- | --- |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 Мбайт |

**Острова.** «Море», в котором есть острова, задается в виде квадратной таблицы размерности N×N. Каждый элемент таблицы равен нулю, либо единице. Под «островом» понимается группа единиц (либо одна единица), со всех сторон окруженная нулями или границами таблицы. Единицы относятся к одному «острову», если из одной из них можно перейти к другой «наступая» на единицы, расположенные в соседних клетках. Соседними являются клетки, граничащие по горизонтали или вертикали. Клетки, граничащие по диагонали не являются соседними. Напишите программу, которая вычисляет количество «островов».

**Формат входных данных**

В первой строке файла записано натуральное число N не больше 100 — размер квадратной таблицы. В следующих N строках задаются элементы таблицы, разделенные пробелом.

**Формат выходных данных**

Выходной файл содержит единственное число — количество островов.

**Пример входных и выходных файлов**

|  |  |
| --- | --- |
| Task3.in | Task3.out |
| 5  1 0 1 1 1  0 0 0 0 0  1 1 1 0 1  0 1 0 0 1  0 0 0 1 1 | 4 |
| 10  0 0 1 1 1 1 1 1 1 1  0 0 1 1 1 1 1 1 1 1  0 0 0 0 1 1 1 1 0 0  0 0 0 0 1 1 1 0 0 1  0 1 1 1 1 1 0 0 1 1  0 0 1 1 1 0 0 1 1 1  0 0 1 1 0 0 1 1 1 1  0 0 0 0 0 1 1 0 0 0  0 0 0 0 0 1 1 1 1 1  0 0 0 0 0 1 1 1 1 1 | 2 |