

Домашняя работа №2

Задание 1. На стандартном потоке ввода задаётся последовательность, описывающая оценки студентов в семестре. Каждый элемент последовательности имеет следующую структуру:

Фамилия Оценка

фамилия представляет собой строку, не содержащую пробельных символов, *оценка* — это целое число от 0 до 5.

Для всех студентов, упомянутых во входном потоке, напечатайте их средний балл. Студенты должны быть перечислены в лексикографическом порядке.

Пример ввода:

```
Clinton 4
Bush 0
Bush 1
Clinton 5
```

Пример вывода:

```
Bush 0.5
Clinton 4.5
```

Задание 2. На стандартном потоке ввода задаётся последовательность, описывающая оценки студентов в семестре. Каждый элемент последовательности имеет следующую структуру:

Фамилия Дата Оценка

фамилия представляет собой строку, не содержащую пробельных символов, *дата* записывается в формате *YYYY/MM/DD*, где *YYYY* — год, *MM* — номер месяца (от 1 до 12), *DD* — день месяца (от 1 до 31). *оценка* — это целое число от 0 до 5.

По введённым данным напечатайте таблицу семестра. В строках таблицы размещаются фамилии студентов, упорядоченные лексикографически (тонкости, связанные с кодировками, не учитывайте). В столбцах размещаются даты в порядке возрастания. Даты должны выводиться в описанном выше формате. В ячейке таблицы размещается символ «точка», если данный студент в данный день не получил никакой оценки, и оценка, если оценка была получена.

Указание. Напишите класс `Date` для работы с датами.

Пример ввода:

```
Clinton 1983/02/1 5
Bush 1976/1/02 0
```

Пример вывода:

```
.          1976/01/02 1983/02/01
Bush      0          .
Clinton .           5
```

Замечание: выравнивать вывод по столбцам не обязательно. Обратите внимание на символ «точка» в левой верхней ячейке таблицы. Номер месяца и дня месяца должен выводиться при необходимости с незначащим нулём (как на примере). Для управления форматированием потока вывода можно использовать *манипуляторы*.

Задание 3. Используя класс `list` решите следующую задачу.

Напишите программу, выполняющую следующие действия. На первом этапе программа считывает со стандартного потока ввода целые положительные числа до тех пор, пока не будет считано число 0 или пока не будет достигнут конец ввода. Считанные числа помещаются в список в порядке их следования во входном файле. Число 0 не является элементом списка. На втором этапе программа выполняет операции редактирования списка согласно командам, считываемым со стандартного потока ввода. Команды считываются до признака конца ввода. Каждая команда состоит из одного или двух целых чисел.

1. Команда, определяемая отрицательным числом x , задаёт, что в списке нужно удалить элемент с номером $-x$. Если в списке нет такого количества элементов, команда игнорируется. Например, команда -2 должна удалить второй элемент от начала элемента списка. Элементы нумеруются от 1. Динамическая память, занимаемая удалённым элементом списка, должна быть освобождена.
2. Команда, определяемая двумя положительными числами x, y , задаёт, что перед элементом с порядковым номером x нужно вставить элемент со значением y . Если в списке нет такого количества элементов, новый элемент добавляется в конец списка. Например, команда $1\ 10$ добавляет перед первым элементом списка (то есть в начало списка) элемент со значением 10.
3. Все прочие варианты являются недопустимыми и в потоке команд не встречаются.

После того, как все команды обработаны, напечатайте получившийся список на стандартный поток вывода в порядке следования его элементов. Каждый элемент списка напечатайте на отдельной строке. После вывода списка динамическая память, занимаемая списком, должна быть освобождена.

Пример входных данных.

```
2 3 4 5 6 7 8 0
1 1 -5 20 9
```

Результат работы программы.

```
1
2
3
4
6
7
8
9
```