

## Простые задачи на работу со структурами

В задачах не предполагается нахождение какого-либо оптимального решения. Решение должно быть правильным, а на его оптимальность можно не обращать внимания.

Формулировки всех задач во многом неформальные (как в реальной жизни). Постарайтесь рассматривать задачи как бытовые и стараться понимать их на этом уровне. Ввод данных, решение задачи, вывод данных должны быть оформлены в виде отдельных функций.

Во всех задачах должна быть корректная очистка памяти в конце работы.

1. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив структур.

Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы). Отсортировать группы студентов по количеству человек в группе по убыванию.

Вывести данные по первым 10% непустых групп (с самым большим количеством студентов) в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).
2. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры.

Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы). Отсортировать группы студентов по среднему рейтингу в группе по убыванию (пустые группы должны находиться в конце массива).

Вывести данные по первым 10% непустых групп (с самым большим средним рейтингом) в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).
3. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив структур.

Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы; указатели на студентов

в структурах Group должны указывать на объекты из исходной структуры *Student\**, т.е. под каждого студента память должна выделяться только один раз). Отсортировать группы студентов по количеству человек в группе по убыванию. Вывести данные по первым 10% непустых групп (с самым большим количеством студентов) в файлы nnn.txt, где nnn – номер группы (т.е. каждая группа выводится в свой отдельный файл).

4. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы; указатели на студентов в структурах Group должны указывать на объекты из исходной структуры *Student\*\**, т.е. под каждого студента память должна выделяться только один раз). Отсортировать группы студентов по среднему рейтингу в группе по убыванию (пустые группы должны находиться в конце массива). Вывести данные по первым 10% непустым группам (с самым большим средним рейтингом) в файлы nnn.txt, где nnn – номер группы (т.е. каждая группа выводится в свой отдельный файл).
5. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив структур.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп, у которых рейтинг ниже заданного рейтинга *R* (индекс равен номеру группы). Отсортировать полученные группы студентов по количеству человек в группе по убыванию.  
Вывести данные по всем полученным непустым группам подряд в файл, разделяя отдельные группы пустой строкой.
6. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп, у которых рейтинг выше заданного рейтинга *R* (индекс равен номеру группы). Отсортировать полученные группы студентов по количеству человек в группе по убыванию.

Вывести данные по всем полученным непустым групп подряд в файл, разделяя отдельные группы пустой строкой.

7. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив структур.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп, у которых рейтинг меньше заданного *R* (индекс равен номеру группы; указатели на студентов в структурах *Group* должны указывать на объекты из исходной структуры *Student\**, т.е. под каждого студента память должна выделяться только один раз). Отсортировать полученные группы студентов по количеству человек в группе по убыванию. Вывести данные по всем непустым группам в один файл с разделителем пустой строкой между группами.
8. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп с рейтингом больше заданного *R* (индекс равен номеру группы; указатели на студентов в структурах *Group* должны указывать на объекты из исходной структуры *Student\*\**, т.е. под каждого студента память должна выделяться только один раз). Отсортировать полученные группы студентов по среднему рейтингу в группе по убыванию (пустые группы должны находиться в конце массива).  
Вывести данные по всем непустым группам в файл с разделителем пустая строка между группами.
9. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив структур.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы). Отсортировать студентов в каждой группе по алфавиту.  
Вывести данные по всем непустым группам в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).

10. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы). Отсортировать студентов в каждой группе по алфавиту.  
Вывести данные по всем непустым группам в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).
  
11. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив структур.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы; указатели на студентов в структурах *Group* должны указывать на объекты из исходной структуры *Student\**, т.е. под каждого студента память должна выделяться только один раз). Отсортировать студентов в каждой группе по рейтингу по возрастанию.  
Вывести данные по первым 10% студентов в каждой непустой группе (с самым маленьким рейтингом) в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).
  
12. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры.  
Создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[1000]* студентами соответствующих групп (индекс равен номеру группы; указатели на студентов в структурах *Group* должны указывать на объекты из исходной структуры *Student\*\**, т.е. под каждого студента память должна выделяться только один раз). Отсортировать студентов в каждой группе по рейтингу по возрастанию.  
Вывести данные по первым 10% студентов в каждой непустой группе (с самым маленьким рейтингом) в отдельный файл с разделителями пустыми строками между группами.
  
13. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить

указанные данные в указанный массив структур. Отсортировать массив студентов по рейтингу.

Требуется объединить студентов в новые группы.

Для этого надо создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[10]* студентами соответствующих групп, заноса студентов из исходного массива в группы подряд (новый номер группы будет равен индексу массива плюс 301). Количество студентов в группах должно быть примерно одинаковым.

Отсортировать группы студентов по максимальному рейтингу в группе в группе по убыванию.

Вывести данные по всем непустым группам в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).

14. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры. Отсортировать массив студентов по номеру группы.

Требуется объединить студентов в новые группы.

Для этого надо создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив студентов данной группы *Student\** и их количество *n*. Заполнить массив групп *Group[10]* студентами соответствующих групп, заноса студентов из исходного массива в группы подряд (новый номер группы будет равен индексу массива плюс 301). Количество студентов в группах должно быть примерно одинаковым.

Отсортировать группы студентов по среднему рейтингу в группе по убыванию (пустые группы должны находиться в конце массива).

Вывести данные по первым 10% непустых групп (с самым большим средним рейтингом) в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).

15. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива структур *Student* создать переменную типа *Student\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив структур. Отсортировать массив студентов по имени.

Требуется объединить студентов в новые группы.

Для этого надо создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[10]* студентами соответствующих групп, заноса студентов из исходного массива в группы подряд (новый номер группы будет равен индексу массива плюс 301). Количество студентов в группах должно быть примерно одинаковым.

Указатели на студентов в структурах *Group* должны указывать на объекты из исходной структуры *Student\**, т.е. под каждого студента память должна выделяться только один раз. Отсортировать группы студентов по минимальному рейтингу в группе по убыванию.

Вывести данные по первым 10% непустых групп (с самым большим количеством студентов) в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая

группа выводится в свой отдельный файл).

16. Создать тип данных *структура Student*, содержащую поля: строка *Name*, целое *Group*, вещественное *Rating*. Для создания массива указателей на структуры *Student* создать переменную типа *Student\*\**. Записать в файл соответствующие данные по студентам (по одному студенту в строке). Загрузить указанные данные в указанный массив указателей на структуры. Отсортировать массив студентов по номеру группы. Требуется объединить студентов в новые группы.  
Для этого надо создать массив групп студентов = *структур типа Group*, каждая из которых содержит массив указателей на студентов данной группы *Student\*\** и их количество *n*. Заполнить массив групп *Group[10]* студентами соответствующих групп, занося студентов из исходного массива в группы подряд (новый номер группы будет равен индексу массива плюс 301). Количество студентов в группах должно быть примерно одинаковым.  
Указатели на студентов в структурах *Group* должны указывать на объекты из исходной структуры *Student\*\**, т.е. под каждого студента память должна выделяться только один раз. Отсортировать группы студентов по среднему рейтингу в группе по убыванию (пустые группы должны находиться в конце массива).  
Вывести данные по всем непустым группам в файлы *nnn.txt*, где *nnn* – номер группы (т.е. каждая группа выводится в свой отдельный файл).