

Оглавление

Требования к домашнему заданию:	2
Оценка домашнего задания:	2
Тестирование домашнего задания:	2
Порядковые номера в журнале и варианты:	2
Описание задания	3
Сложность 1	3
Файл «books.csv»	3
Файл «students.csv»	4
Сложность 2	5
Сложность 3	6
Бонус. Дополнительная функциональность (для всех сложностей и вариантов):	6

Требования к домашнему заданию:

1. Код пишется **только** на языке программирования C.
2. Можно использовать стандартные библиотеки языка C.
3. Использовать чужой (списанный у одногруппников) код запрещается.
4. Готовность ответить на любой вопрос по коду (зачем та или иная переменная, как работает та или иная функция)

Оценка домашнего задания:

1. Сложность 1 – 10 баллов
2. Сложность 2 – 15 баллов
3. Сложность 3 – 20 баллов
4. Использование подходящих алгоритмов (быстрая сортировка, сортировка Шелла, двоичный поиск) и структур данных (стэк, связный список, двоичное дерево, хэш-таблица) - + 5 баллов вне зависимости от сложности.
5. «Правильное» оформление кода (использование функций, переменные с нормальными названиями, отступы, отсутствие дублирования кода (copy/paste вместо выделение функций), понятные сообщения об ошибках и т.п.) - +5 баллов вне зависимости от сложности.
6. Если ДЗ не получается защитить (ответить на вопросы преподавателя) оценка снижается на 5 баллов.
7. Наличие списанного кода – оценка снижается по усмотрению преподавателя.

Тестирование домашнего задания:

1. Все функции, которые указаны в соответствующих вариантах должны работать корректно
2. Программа должна работать с тестовыми файлами преподавателя.

Порядковые номера в журнале и варианты:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	x						x						x						x					
2		x						x						x						x				
3			x						x						x						x			
4				x						x						x						x		
5					x						x						x						x	
6						x						x						x						x

Описание задания

База данных университетской библиотеки состоит из нескольких файлов в формате «.csv» (описание формата <https://ru.wikipedia.org/wiki/CSV>).

Сложность 1

Файл «books.csv»

Первый файл «books.csv» содержит информацию о всех книгах в университетской библиотеке. Каждая запись включает номер ISBN книги (уникальный номер), ФИО авторов, название книги, количество книг всего, количество доступных книг.

Пример:

9666965348;Стругацкие;Град обреченный;10;5

5845908914;«Керниган, Ритчи»;Язык программирования C;13;10

Т.е. в библиотеке всего есть 10 книг «Град обреченный» и 5 из них у студентов.

Напишите программу, которая считывает файл с данными по книгам и выводит меню с доступными операциями:

1. Добавить новую книгу (при добавлении проверять нет ли уже такой книги по номеру ISBN)
2. Удалить книгу по номеру ISBN
3. Просмотр всей информации по книге (здесь и далее книга выбирается по номеру ISBN, если введен неправильный номер выводится сообщение о том, что книга не найдена).
4. Вывести информацию по всем книгам в виде таблицы, записи должны быть отсортированы по номеру ISBN
5. Редактировать информацию по книге.
6. Изменить количество книг в библиотеке по номеру ISBN (необходимо пересчитать количество доступных книг).
7. Выдать книгу студенту по номеру ISBN (необходимо уменьшить количество доступных книг на 1, если количество доступных книг – 0, вывести соответствующее сообщение).
8. Принять книгу от студента по номеру ISBN (необходимо увеличить количество доступных книг на 1, если количество доступных книг равно количеству книг в библиотеке, вывести соответствующее сообщение).

9. Сделать бэкап (сохранить состояние программы в файл «.csv», «books_<дата и время>.csv», вывести сообщение успешно или не успешна завершилась операция).
10. Восстановить базу из файла бэкапа (имя файла бэкапа указывает пользователь, вывести сообщение как прошло восстановление данных).
11. Поиск по фамилии автору (пользователь вводит фамилию автора, если в библиотеки есть одна или несколько книг автора – выводится вся информация по ним, если таких книг нет, выводится сообщение, что книг не найдено)
12. Завершить работу программы (текущее состояния библиотеки сохраняется в файл).

Варианты:

1. Функции меню: 1, 2, 3, 4, 12
2. Функции меню: 1, 2, 5, 6, 12
3. Функции меню: 1, 2, 7, 8, 12
4. Функции меню: 1, 2, 9, 10, 11, 12

Файл «students.csv»

Второй файл «students.csv» содержит информацию о студентах: номер зачетной книжки, фамилия, имя, отчество, факультет, специальность.

1K2342;Иванов;Иван;Иванович;ИУ;«Проектирование и технология производства ЭС»

Напишите программу, которая считывает файл с данными по студентам и выводит меню с доступными операциями:

1. Добавить студента (при добавлении проверять, что такого студента еще нет по номеру зачетной книжки)
2. Удалить студента по номеру зачетной книжки
3. Редактировать информацию по студенту
4. Просмотреть информацию по студенту по номеру зачетной книжки
5. Сделать бэкап (сохранить состояние программы в файл «.csv», имя файла «students_<дата и время>.csv», вывести сообщение успешно или не успешна завершилась операция).
6. Восстановить базу из файла бэкапа (имя файла бэкапа указывает пользователь, вывести сообщение как прошло восстановление данных).
7. Поиск по фамилии студента (пользователь вводит фамилию студента, если библиотекой пользуется один или несколько студентов-однофамильцев – выводится вся информация по ним, если студента нет – выводится соответствующее сообщение)
8. Завершить работу программы (текущее состояния библиотеки сохраняется в файл).

Варианты:

5. Функции меню: 1, 2, 3, 4, 8
6. Функции меню: 1, 2, 5, 6, 7, 8

Сложность 2

В третьем файле «users.csv» хранится информация о пользователях системы в формате: логин, пароль, просмотр и редактирование информации по студентам, просмотр и редактирование информации по книгам. Пример:

```
admin;qwerty;1;1
```

```
user1;zxcasd;1;0
```

```
user2;zxcasd;0;1
```

Флаги 1 и 1 у пользователя admin означают, что он имеет доступ к просмотру и редактированию информации как по книгам, так и по студентам.

Напишите программу, которая считывает файл с данными по пользователям, книгам и студентам и в которой реализованы следующие функции:

1. При запуске программы требуется ввести логин и пароль, если они указаны неверно выводится ошибка.
2. Если пользователь имеет доступ только к просмотру и редактированию книг выводить меню для книг (см. описание первого файла)
3. Если пользователь имеет доступ только просмотру и редактированию выводить меню для студентов (см. описание второго файла)
4. Если пользователь имеет доступ и к книгам, и к студентам – выводить меню верхнего уровня: 1 – Книги, 2 – Студенты. В меню книг и студентов добавить функцию: 0 – возврат в основное меню.

Варианты

1. Все функции для авторизации (1, 2, 3, 4), задания для варианта 1 и 5 из сложности 1.
2. Все функции для авторизации (1, 2, 3, 4), задания для варианта 2 и 5 из сложности 1.
3. Все функции для авторизации (1, 2, 3, 4), задания для варианта 3 и 6 из сложности 1.
4. Все функции для авторизации (1, 2, 3, 4), задания для варианта 4 и 6 из сложности 1.
5. Все функции для авторизации (1, 2, 3, 4), задания для варианта 3 и 5 из сложности 1.
6. Все функции для авторизации (1, 2, 3, 4), задания для варианта 1 и 6 из сложности 1.

Сложность 3

В четвертом файле «student_books.csv» содержится информация о книгах, которые сейчас у студентов: ISBN книги, номер зачетки студента; дата возврата книги (дд.мм.гггг). Пример: 9666965348;1K2342;«01.06.2015»

Добавляются следующие функции:

1. Нельзя удалить студента, у которого есть книги.
2. Нельзя удалить книгу, которая есть у студентов.
3. Нельзя изменять ISBN книг и номер зачетки студентов.
4. Для каждого студента (по номеру зачетки) есть возможность посмотреть информацию по всем его книгам (информация из книги и дата сдачи в библиотеку).
5. Для каждой книги можно посмотреть информацию по студентам, которые ее взяли (информация о студенте и дата сдачи в библиотеку).
6. При выдаче и сдаче книги студенту пользователь должен указать ISBN книги и номер зачетки студента.
7. Если нет доступных книг для выдачи, необходимо выводить сообщение – когда будет сдана ближайшая книга.

Вариантов нет, необходимо реализовать все функции.

Бонус. Дополнительная функциональность (для всех сложностей и вариантов):

В пятом файле «library.log» ведется лог действий пользователей в системе в формате дата и время действия, логин пользователя (для сложности 1 логин не нужен), название функции из меню (см. описание функций файлов 1, 2). Пример:

“10.05.2015 21:00:12”;“admin”;“deleteBook”

Так же в лог можно добавить:

- значения параметров функций при их вызовах
- все сообщения, которые выводятся пользователю: ошибки и подтверждения успешного завершения той или иной функции.