

FILE ORGANIZATION LAB PROJECT

A librarian has 500 books all numbered from 0 to 499 as book id numbers and a bookshelf that has 30 shelves. Each shelf can contain 5 books. He/She wants to keep a ledger on his computer. He/She needs you!!!!

So you will use bucket hashing to index the books. You have to control two files.(you may use sequential or random access as you wish) First file contains the index data, the example is shown below: (you may select any hashing function possible) (First column is Shelf number and addition 5 columns reserved for 5 book slots for each shelf. Each one of those are book ids replaced to slots by your hashing function. **Don't forget to inform me what is your hashing function**)

Shelf number	B1	B2	B3	B4	B5
1	31	45	58	NULL	NULL
2	63			
3				
..					
30				

Examples throughout this project's announcement given only for your information, the datas are absolutely random!!!!

The second file will hold author and book name for example: (each line shows the appropriate shelf number and each line has 5 author name and book name,if there isn't any book in one slot of the shelf it must be shown as NULL in the right place of the file. If the appropriate slot is a tombstone "at least an item is registered there and deleted afterwards" you need to mark the slot with 3 slashes "///")

Author name	Book Name	Author name	Book Name	Author name	Book Name...
Tolkien	The Hobbit	///	///	NULL	NULL
.....					
.....

In the example above there is only 1 book on the shelf, one is a tombstone and the other is marked NULL.

The keys must be inserted from the keyboard by the user and your program must place it into the index file and must ask the books author name and book name to user and register it into the second file named as **books**. Index file name is up to you. If the files weren't created , your application must create it. If the file is present it will add the new inserts.

Also there must be a search step for users to find any books place,name and author name by giving anyone of the book datas(book number, name or author name)

Important Note: As you see there is 500 books but 150 slots to insert. If 5 slots for any shelf is taken and there is a 6th book to insert, you will not insert it and alert the user that there is no place.

So your application must have 5 menu items:

1-Insert a book (you need to ask book id ,book author and book name to the user , book id will be hashed, and all information will be recorded to the needed files. If the resulted slot is a tombstone, you will print a message "This slot was a tombstone")

2-Search a book (there must be 3 criteria to search, book id, book name and book author, user will select one of them.You can create a sub menu to implement this. Also you need to count and list how many search you made and how many tombstones you trespassed)

3- Delete a book (there must be 3 criteria to search, book id, book name and book author, user will select one of them.You can create a sub menu to implement this. By deleting you have to mark the appropriate slots as tombstone "///")

4-List all the bookshelf (Look below for further instructions)

5-Exit (Finish the application)

After the user selected the menu items 1-2-3 or 4, you must stop the screen flow in order to enable the user to see the result of his/her choosing, after user has done his work enable the user to press a button to make the main menu reappear to provide further selection.

P.S.All changes about books must be done on both ".dat" files.

Listing all the bookshelf will list all books only by **book name** inserted like the format below:

Shelf number	B1	B2	B3	B4	B5
1	Hobbit	Call of the Wild	NULL	NULL	///
				
30				

The format and rules:

- 1) Every student will do his/her assignment alone. (even if they are siblings....)
- 2) You name your c file as your student number.c (ex:1306120001.c)
- 3) You will send as attachment the c file ,index file ,books file crunched as zip or rar file named "student number-Student name.zip" like "1306120001-Elif Türk.zip" The zip or rar file contains for example : 1306120001.c, index.dat,books.dat files.
- 4) Project must be sent to the course's Google classroom before the deadline.

- 5) Score will be diminished if you don't follow the rules. You will get **0** if you don't follow rule number 1, you will also get 0 if you pass the deadline.
- 6) **The deadline is 12 May 2019 - 23:59.**

Good Luck.....

Dr.M.Deniz DEMİRCİ