File Structure: Assignment #3

100points

Cairo University, Faculty of Computers and Information

Notes:

1. *Cheaters will be graded by –ve points , Don't copy any code from anywhere ..*
2. *Submit your code to through Acadox only.*
3. *Submit one compressed file with you IDs and Group Name*
4. *Due Date **13/5/2018 11:30 PM***
5. *Team = **max 3 students**, team must be from the same lab*

Consider we want to store data about students, each student record has the data below.

- ID: max char[9]
- Name: max char[49]
- Address: max char[49]

Problem #1 you are required to search data inside the data file using a primary index using the ID field

- Implement the primary index using binary search tree (AVL tree is recommended)
- Data file organization is (variable length record): length indicator records, delimited fields.

Problem #2 you are required to search data inside the data file using the hashing technique.

Implementation details:

- Address space **N= 997**.
- Bucket size =2.
- Hashing method: Use *Chaining with separate overflow area* technique to handle collision, The hash function is the *multiplication method*
- Data file organization is: fixed record length (record fixed length is 110 chars), delimited fields
- You can use auxiliary data structures in RAM or not.

Problem #3 Now suppose that the student's record is variable length record, not fixed, and the student's record data has the same above fields, you are required to search inside the data file using the following hashing technique:

- The primary Index is Hashed (index size N= 997) (Note: check lecture 15 for more details)
- Bucket size =1.
- Data file organization is (variable length record): length indicator records, delimited fields.
- The hash function is the *multiplication method*

Testing

1. You are required to create **20** random records;
2. for every problem of the three problem above
   i. insert the **20** records in the data file.
   ii. **print** all inserted records.
   iii. Delete 5 random records from the file.
   iv. Print the remaining 15 records after deletion.

Implementation details

- *The three problems are totally separate problems.*
- *The random key should be unique.*
- No need to use avail list, mark the deleted record by *
- If you want to use avail list, it's OK. No need to use a status flag to check index integrity.
- ***Assume any other information you need.***