File Management and Processing

Cairo University, Faculty of Computers and Information

Delivery Instruction:

1. Cheaters will be graded by *–ve points, Don't copy any code from anywhere.*
2. Due Date: **Week starting December 3, 2022**
3. Team = max 4 students.
4. No late submission will be accepted.
5. The assignment weights 10 grades.

We want to store data about **Employees** and **Department**. (for simplicity consider any employee belongs to only one department, but any department may have more than one employee)

Employee attributes	**Department attributes**
Char [13]: Employee_ID //primary key Char [30]: Dept_ID //sec. key Char[50]: Employee _Name Char[50]: Employee _Position	Char [30]: Dept_ID //primary key Char[50]: Dept_Name //sec. key Char[50]: Dept_Manger

Consider we want to save 10 Employees and 5 Department.

Save the data for employees and department in the following format: **delimited fields, length indicator records.**

You should write the producers that build the following indexes:
1. Primary index using the Employee_ID (for Employees datafile)
2. Primary index using the Dept_ID (for Department datafile)
3. Secondary index using Dept_ID (for Employee datafile) //Dept _ID is sec. key in Employees datafile
4. Secondary index using Dept_Name (for Department datafile)

The user can write a query that contains **fixed** key words (formatted in **red** below)

Examples for queries that user can write

▪ **select all from** Employee **where Dept_ID =** 'xxxx' // this query will use sec. index to get results
▪ **select all from** Department **where Dept_ID =** 'xxxx' // this query will use primary. index to get results
▪ **select** Employee_Name **from** Employee **where Employee_ID =** 'xxxx' // this query will use sec. index to get results

File Management and Processing

The main welcome screen is below.

- Add New Employee
- Add New Department
- Delete Employee (ID)
- Delete Department (ID)
- Print Employee (ID)
- Print Employee (Dept_ID)
- Print Department (ID)
- Print Department (name)
- ~~Write a Query~~
- Exit

Important notes:

All indexes are sorted ascending

You MUST implement secondary indexes using Linked List *technique.*

Searching in indexes is performed using **binary search**.

To delete a record just put an * in the beginning of that record and add this record to avail list.

All operations (add, delete) will affect indexes.

Search operations will use indexes (primary or secondary)

Bind all secondary indexes with the primary index, don't bind them by addresses directly.