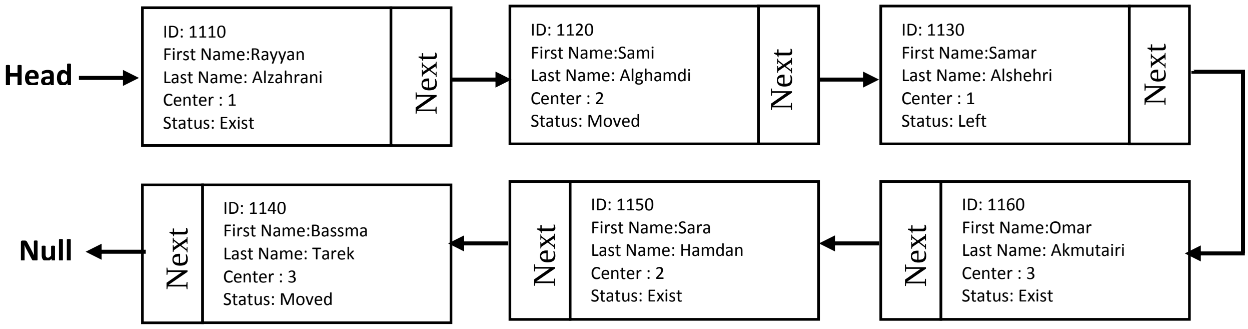
# Concept Application & Algorithmic Part

**Background:**

At KAU, there is many vaccination centers which distribute in different centers. The capacity of these centers is different where the distribution of health practitioners to give vaccine doses over the available centers is based on the capcity of each center. The responsible management of vaccine determine the number of existing centers at KAU and the capacity of each center. The Practitioners can move to aonther center or to leave the center. Your task is to write a program to do the task of distribution of of health practitioners over the available centers on behalf of the Vaccination management.

*Question 1:* (20 points)

1. **Concept Application:** Suppose a linked list contains the following information about the health practitioner. Write the steps to delete the nodes of **"Left" status**. Your answer must indicate:
   - The **position of the pointer** and
   - The status of the **linked list** after every step.

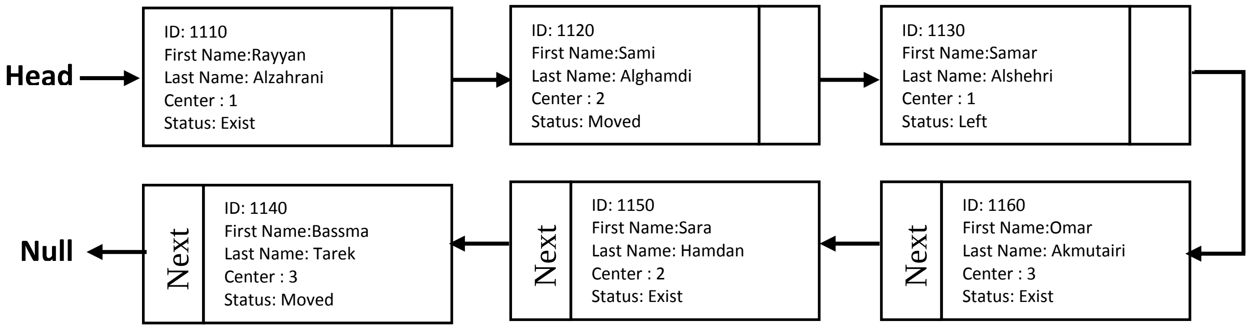


*Question 2:*(20 points)

2. **Algorithm Write up**: Suppose a **linked list** contains all health practitioners' information for all centers. The vaccination management employee wants to display all information of health practitioners in any required center.

**Write an algorithm** that display all information of health practitioners in the linked list for any required center.

For example, an employee of vaccination management would like to display all information of health practitioners of center 2. If the linked list contains the following nodes



**The output of the algorithm should print**

```
The information of health practitioners in center 2 are:

1: ID: 1120, Name: Sami Alghamdi Status: Moved.
2: ID: 1150, Name: Sara Hamdan   Status: Exist.
```

**Algorithm**
**Input:**

**Output:**

**Method:**

*Question 3:*(60 points)

## Program 2: KAU Vaccination Centers Management System

### Objective

The primary objective of this program is to implement a linked list. The secondary objective is to practice with File I/O and arrays.

### Program Description

Write a program to manage the distribution of health practitioners among the vaccination centers. You are required to write the following important methods:

1. Write a function to add a practitioner to the specified center (**at the end of the linked list**).
2. Write a function to search for the practitioner based on his id.
3. Write a function that removes the left practitioner for specified center (linked list) into a new linked list that represents the left practitioners.
4. Write a function that deletes the node from the linked list based on the practitioner id.

The program deals with three files. Two input files and one output file. The description of these files as follow:

- The first input file (***intialInformation.txt***) contains the important information for the system which includes the number of centers, capacities of each center, and practitioners information. The information in this file are arranged as follow:
    - The first line contains **number of centers, and the capacities of all centers.**
    - The second line contains the **centers names**
    - The remaining lines contain the health pratitioners information**(id, first name, last name)**

- The commands for the system are found in the second file called ***commands.txt.*** The commands in this file as follow:
    - **STARTUP**: This command will use the first input file (***intialInformation.txt***) to initialize the system by creating numbers of linked lists equal to the number of centers specified in the file (every linked list holds the health practitioners of one center). Then, it will distribute the health practitioners among the created lists based on capacities specified in the first line which started from the second number on the line.
    - **DISPLAY_ALL_CENTERS:** This command will display the health practitioners in all centers similar to the output in the ***ouptput.txt*** file
    - **NUM_PRACTIONERS:** This command has one value that determines the center number. It will display the number of health practitioners in the specified center.
    - **DISPLAY:** This command has one value that determines the center number. It will display all health practitioners in the specified center.
    - **DISPLAY_ALL_BASED_ON_STATUS:** This command will display the health practitioners in all centers based on the required status (Exist, Moved, Left). If the required status in not found it will display the "**Not found any practitioners of the status <status>**" message.
    - **DISPLAY_BASED_ON_STATUS:** This command will display the health practitioners in specific center based on the required status (Exist, Moved,

Left). If the required status in not found it will display the "**Not found any practitioners of the status <status> in center <number>**" message.

  - **LEAVE_THE_JOB:** This command has one value that represents the practitioner id who will leave the center. The command will search for this practitioner in all linked lists and change him/his status to ***Left.***
  - **REMOVE_ALL_LEFT_PRACTITIONERS:** This command will delete all left practitioners from all centers (linked lists) and add them to the new linked list that contains all the left practitioners.
  - **MOVE:** This command has two values, the first value represents the practitioner id to be moved and the second value represents the required center for moving. The command will search for this practitioner in all linked lists, remove it from the current center(linked list), add him/her to the required center and change him/his status to ***Moved.***
  - **DELETE_CENTER:** This command has one value that determines the center number. It will delete the whole specified center.
  - **MERGE:** This command will merge all the remainingg linked lists to a new one.
  - **QUIT:** This command will stop the program.

- The output of the program should be written to the file name it ***output.txt,*** which content should be similar to the contents of the file provided to you.

## Implementation

For this program, you will create the following classes:

- *Practitioner.java*: This class will be used to create objects of type practitioner. Each practitioner object will store the practitioner id, practitioner first name, practitioner last name, center number, status (Exist, Moved, Left).

- *Center.java*: All the methods will be implemented in this class.

- *MainProgram.java*: This is the class that will contain the main.

## Sample Input & Output File

We have provided you a sample for two input files and one output file.

*****WARNING*****

Your program MUST adhere to the EXACT format shown in the sample output file (spacing capitalization, use of dollar signs, periods, punctuation, etc). The graders will use large input files, resulting in large output files. As such, the graders will use text comparison programs to compare your output to the correct output. If, for example, you have two spaces between in the output when there should be only one space, this will show up as an error even though you may have the program correct. You will get points off if this is the case, which is why this is being explained in detail. The minimum deduction will be 10% of the grade, as the graders will be forced to go to the text editing of your program in order to give you an accurate grade. Again, your output MUST ADHERE EXACTLY to the sample output.

## Grading Details

Your program will be graded upon the following criteria:

1) Adhering to the implementation specifications listed on this write-up.
2) Your algorithmic design.
3) Correctness.
4) **Use of the three classes, as specified. If your program is missing these elements, you will lose marks.**
5) The frequency and utility of the comments in the code, as well as the use of white space for easy readability. (If your code is poorly commented and spaced and works perfectly, you could earn as low as 80-85% on it.)
6) Compatibility to the **newest version** of NetBeans. (If your program does not compile in NetBeans, you will get a large deduction from your grade.)
7) Your program should include a header comment with the following information: your name, **email**, account number, section number, assignment title, and date.
8) Your output MUST adhere to the EXACT output format shown in the sample output file.

## Deliverables

You should submit a zip file with four files inside:

1. *ConceptPart.doc (Containing concept and algorithm parts)*
2. *Practioner.java*
3. *Center.java*
4. *MainProgram.java*

***These three files should all be INSIDE the same package called **StudentsDistribution**. If they are not in this specific package, you will lose points.

**NOTE: your name, ID, section number AND EMAIL should be included as comments in all files!**

**UML Diagrams:**

For this program, you will create **three** Classes (UML diagram shown below):

**Practitioner**

*Data Members*

```
private String parctID;
private String Fname;
private String lName;
private String Status;
private int center;
private Practitioner next;
```

*Operations/Methods*

`Practitioner () //` one or more Constructors

And any other methods you need.

**Center**

*Data Members*

```
private int centerID;
private Practitioner head;
```

*Operations/Methods*

```
addPractitioner()
searchByID()
deletePractitionersBasedOnStatus()
deletePractitionerById()
```

ALL necessary methods for linked-list operations

**MainProgram**

*Data Members*

As needed

*Operations/Methods*

```
public static void main()
```

ALL necessary methods for all operations