**ICT 201**

Project # 2: Student Plan Cellphone Application Center

For this project, you are to simulate an application process for students applying for a cell phone plan. The application process contains three stages: apply, line up and serve. Apply is the stage where clients will go online to “fill out” an application form containing their personal information. Once they’re done filling out an application form, they line up in a processing center and wait to be served. *Take note that the order in which they apply need not necessarily be the order in which they line up.* Once they are served, they receive a new SIM card with their new cell phone number.

To make this project work, you are to come up with at least one program file called ***CellphoneAppCenter*** which should contain the main method implemented in whatever language you choose. This class should take in three command line arguments: (1) name of input file to read, (2) name of a file containing available cell phone numbers, and (3) name of output file to write to. Feel free to come up with as many classes as you find appropriate for your system. Also, you should be using custom classes for the data structures you will be using for this project and not use the libraries already defined in the language that you are using.

# Description of the System

Just like the old project, you are to get input from a file that dictates what action you are to take on a particular piece of information. Your program should process the input data accordingly and write the output to a file following a prescribed format. Here is a list of actions and expected input/output:

**1) apply info:***idnumber*:*lastname*:*firstname*:*address*:*birthday*

For this command, you are to store the information into a record that uses *idnumber* as a key and the rest as items in specific fields. To keep things simple, *idnumber* should be stored as *int* while *lastname*, *firstname*, *address*, and *birthday* should be stored as *String*. Each application information entry should be stored in a Binary-Search-Tree based Dictionary data structure that uses *idnumber* as a key and the rest of the information as records kept in that node.

A priority number, called *priority* should be generated internally during the apply stage and stored with the record. The priority number is generated sequentially; i.e., the first who applies gets priority number 1, the second who applies gets priority number 2, and so on. This would determine the priority a record should get when it waits to be served after lining up.

This command should output the following information:

**applicant** *idnumber***:** *firstname lastname*

**2) lineup:***idnumber*

For this command, you are to insert the specified element into a priority queue structure. This time, you are to extract *priority* from the record and use this as a key for the priority queue structure. The priority queue structure should also store *idnumber* to be used later for lookup. Implement the priority queue structure using a heap.

This command should output the following information:

**lineup** *idnumber***:** *priority*

**3) serve**

This command determines who gets served first. When a serve command is read, it removes the record with the lowest priority number from the list and looks up its record from the Binary Tree structure. With this, a predetermined phone number, called *cpnumber*, is taken from a list and assigned on a first-come-first-serve basis. The list will come from the text file indicated in the command line.

This command should output the following information:

**Assign mobile #** *cpnumber***, to applicant** *idnumber***:***firstname lastname* **of** *address***,** **born** *birthday***.**

Sample input and output files are provided for your reference.