**Assignment 2 Report**

**1.Problem：**

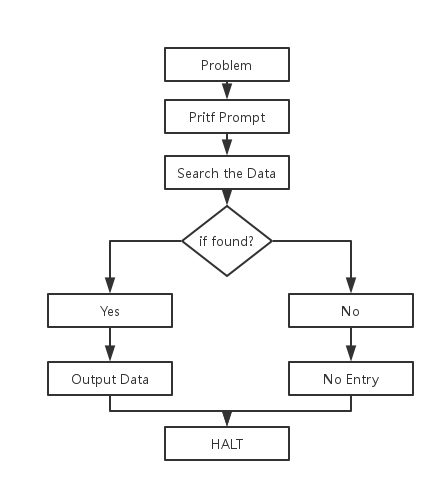
Write a program starting at location x3000 in LC-3 assembly language that will:

1. Upon starting: Print “Type a name and press Enter:” on the screen and then wait for user to input a string. The input will be ended by pressing <Enter> (ASCII code x000A or x000D depending on your operating system). Each name is a case-sensitive string of length between 1 to 15 characters.

2. After the user pressed the <Enter> key, your program will search the directory to find the room number based on that name. The address of the first node of the directory has been already stored in location x4000.

3. For each faculty whose first name **or** last name is the same as input, print his/her full-name and room number in one row. If none exists, print “No Entry”. Each line should be terminated by a carriage return (ASCII x0A).

4. Clean up: Halt the machine.



**2.Main Idea:**

**Input:**

1.Store the character of name inputted in a block of memory

2.Change the value of characters to 2’s complement and store them as a string with the end of ‘Enter’or change them to ‘\0’with the value of 0.

(1)’Enter’: add the value of the ASCII of ‘Enter’to its 2’s complement and test if the user finish input

(2) Or we can change as ‘\0’after the user finish input, and use BR instruction to test if it’s the end of the NAME string

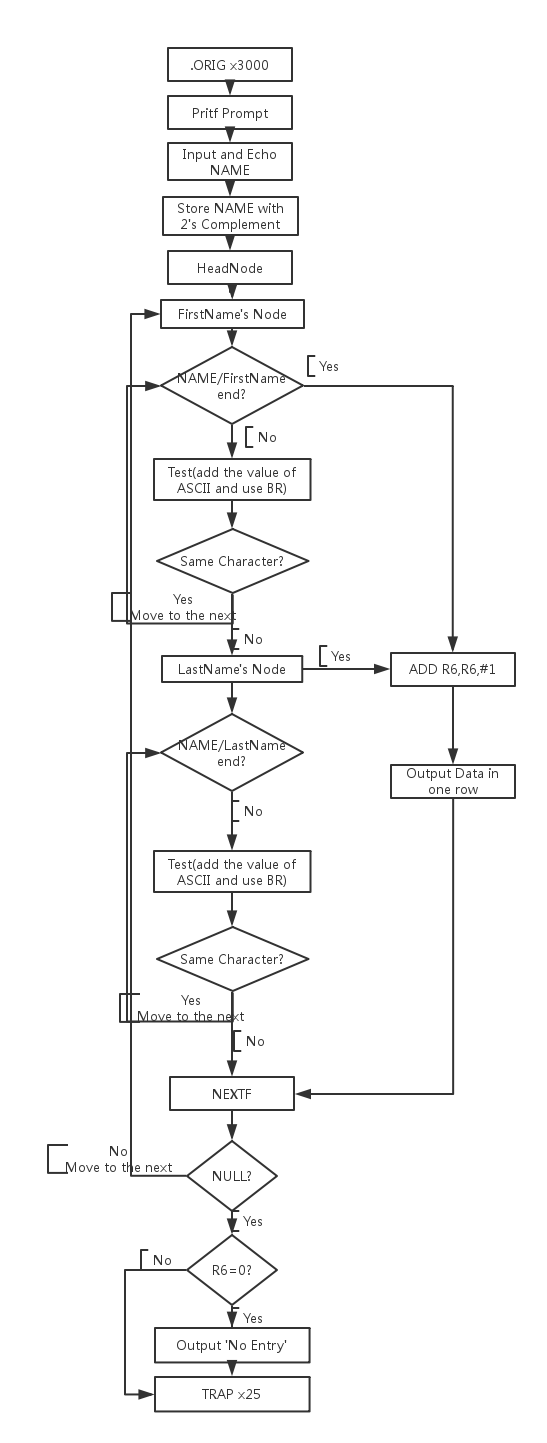
**Search:**

1. The order to Search:

Headnode🡪Node1(Firstname’s node🡪Lastname’s node)🡪Node2(…)🡪……🡪NodeN🡪NULL

**Output:**

1. When NextNode is NULL, that’s means program is performing to the end of the directory.
2. R6 != 0, there is at least one name found in the directory same as the input NAME.
3. R6 = 0, none is found in the directory same as the input NAME. Output ‘No Entry’to inform the user.



**3.Thinking：**

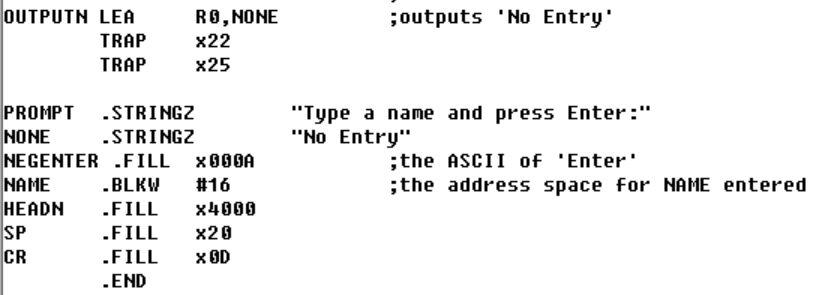
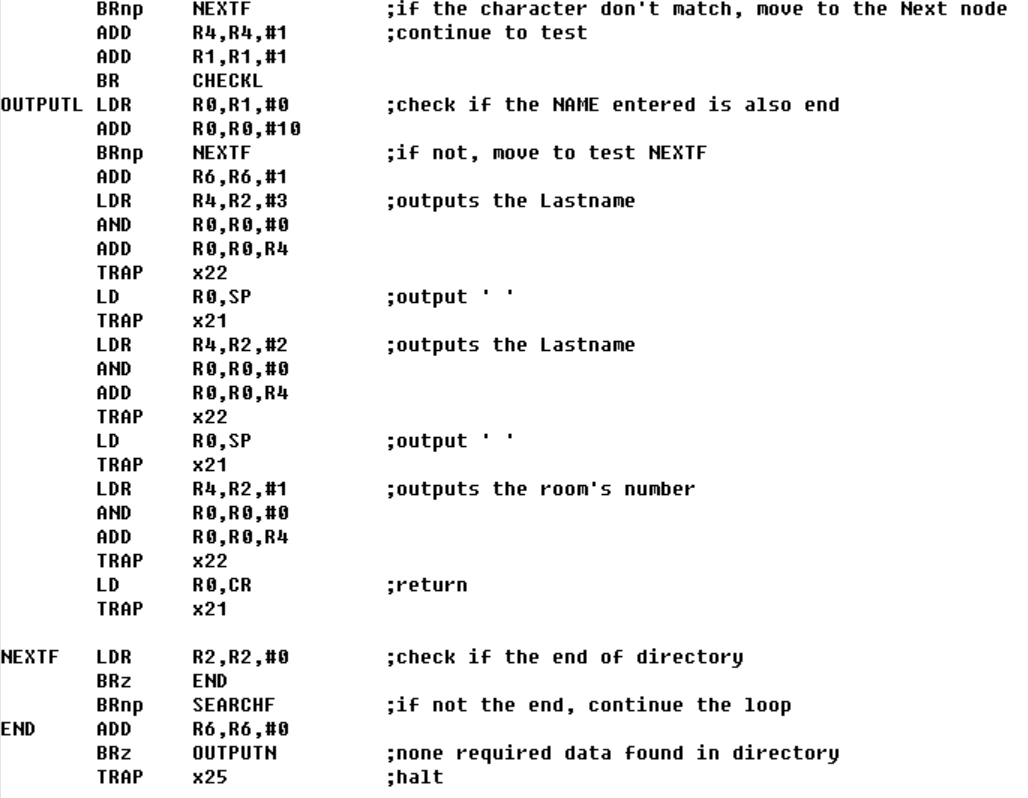
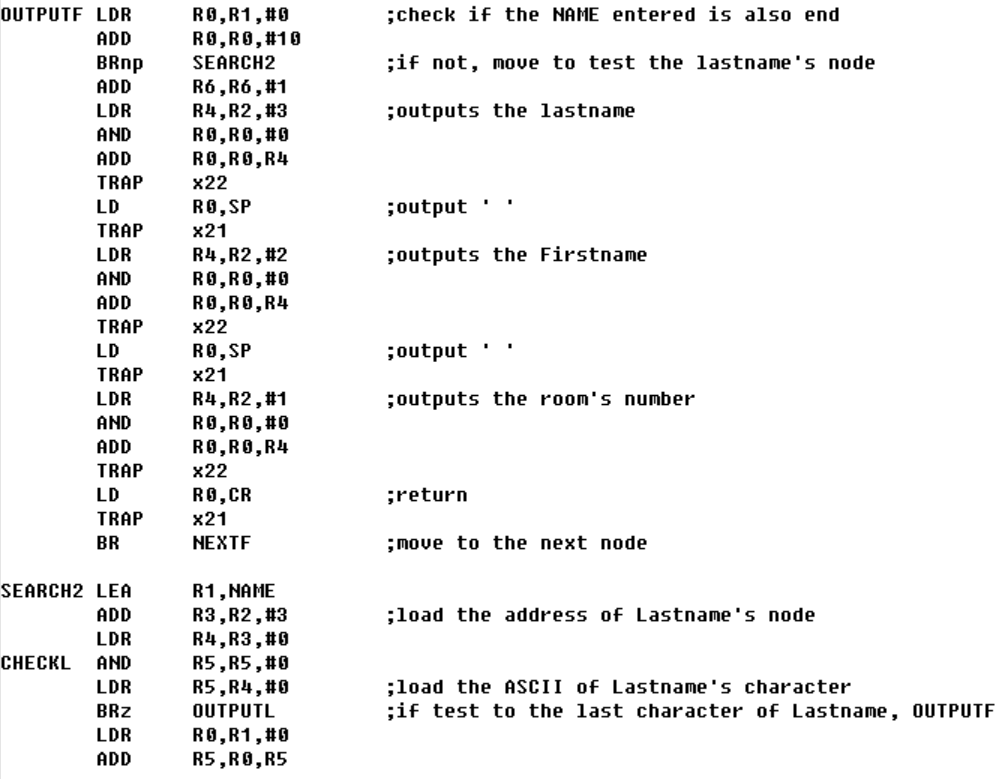
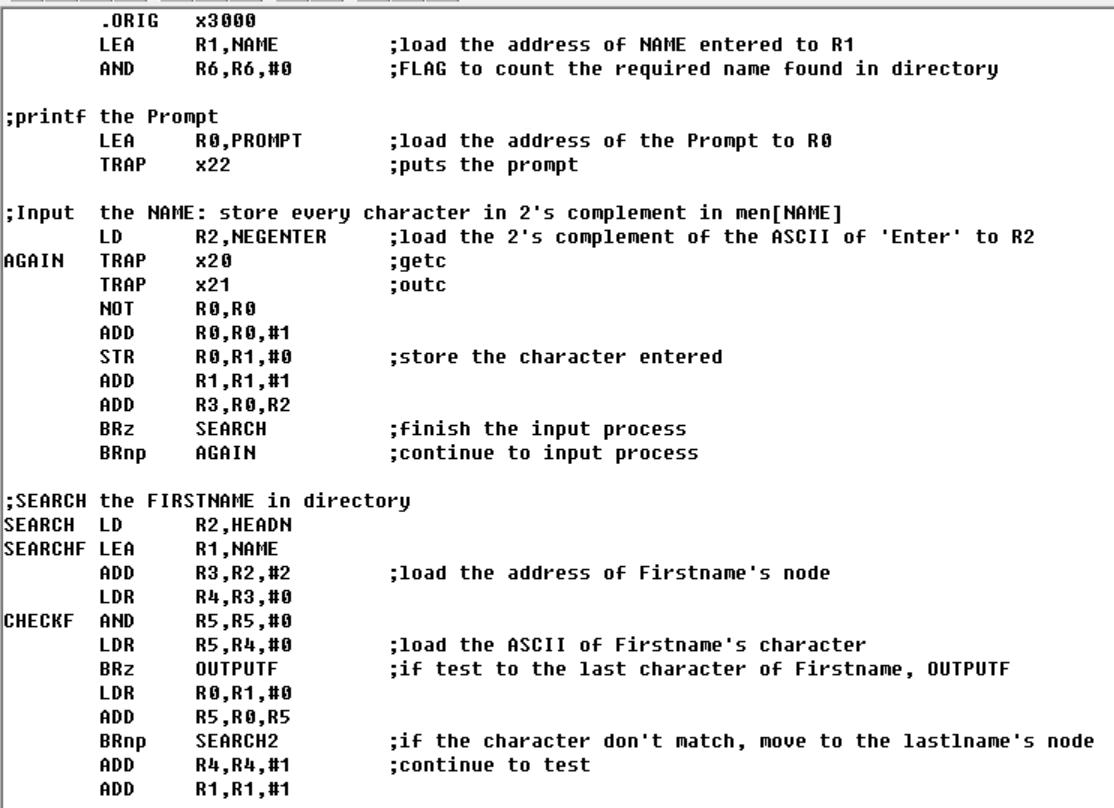
1. The order to search: If the program found the same first name exists in the directory, it should output the data required, add R6 to 1 and jump to the NEXTF. Otherwise, if we have the test case with same first name and last name, the program will output the repeated data twice.

2. Every time finish outputting the data, it should output a return so that the result can satisfy the requirement that every data in one row.

3. There should be a counter (R6) to count the total number of required data so that in the end, the program can test if there is none required data and output ‘No Entry’.

4. In this program, we can regard NAME and characters stored in memory both as strings. Therefore, it’s necessary to test both the end of the string when doing research. Otherwise, if there is test case such as NAME: Jinn and LastName: Jin, BUG!

**4.Program：**



**5.Result：**

