PartA. Algorithm

My code has these algorithms: Character Strings, String Compare and Linked Lists.

Data are stored as Linked List. In each node of the Linked List, there are three pointer to the character strings.

Each node of Linked List consists of:

①The pointer for the next node

②The pointer to an ASCII string representing the room number

③The pointer to an ASCII string representing the first name

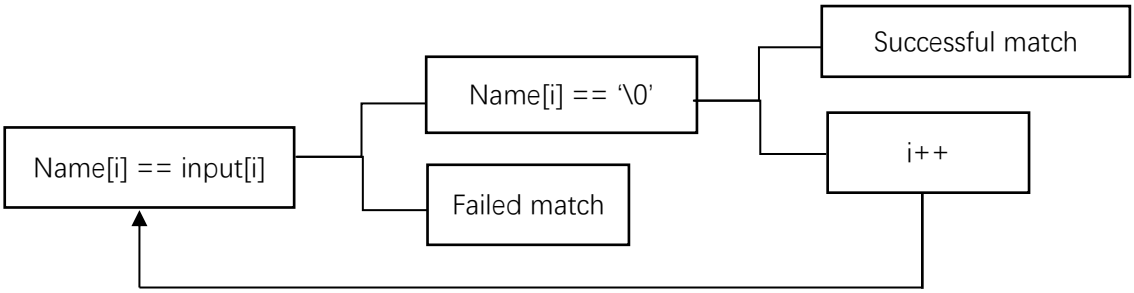
④The pointer to an ASCII string representing the last name

If the pointer for the next node is x0000, the current node is the last node of the Linked List.

To accomplish the task, program iterate over the Linked List, compare the name and the input String, using the String Compare algorithm. I compare first name first. If the first name is same as the input String, then I print this node and jump to the next node. If not, I compare the last name.

To iterate over the Linked List, I use LDR R1,R1,#0 (the current location is stored in R1) to get the next location.

String Compare Algorithm does two things: ①check Name[i] == Input String[i] ②check Name[i] == '\0'



PartB. Essential parts of my code

1.code for input

```
Input          LEA       R1,Name
               LD        R2,NewLine    ;get x-000A
               Not       R2,R2
               ADD       R2,R2,#1
Loop_Input     TRAP      X20
               TRAP      X21
               STR       R0,R1,#0
               ADD       R0,R0,R2
               BRz       EndInput      ;if char is 'x000A', input has finished
               ADD       R1,R1,#1
               BRnzp     Loop_Input
EndInput       STR       R0,R1,#0      ;change the 'x000A' at the end of the STRING to x0000
```

When the user types, the program first check whether it is 'x000A' (start a new line). If not ,put

it in the right location. If so, change the ‘x000A’ to ‘x0000‘.

2. go over the linked list

```
Iterate      LDI      R1,Head
             BRz      Exit              ;check if the linked list is empty
             AND      R3,R3,#0          ;use R3 to store the number of target nodes
Loop_iter    JSR      CheckNode
             Add      R5,R5,#0          ;use R5 to store the return value(1-match; 0-not match)
             BRp      PrintNode
Continue     LDR      R1,R1,#0          ;check if the next node is x000
             BRz      Exit
             BRnzp    Loop_iter
;
Exit         ADD      R3,R3,#0
             BRp      ExitAll           ;if R3 == 0, there is no node matched, print "Not found"
             LEA      R0,NF
             TRAP     X22
ExitAll      TRAP     X25
```

Use LDR to get the next location until it is ‘x0000’

PartC. The questions that TA asked you, and answers.

1.How to iterate over linked list:

①use LDI to get the first location. Check if its address is x0000, if so, exit

②use LDR to get next locations until the next location is x0000, use BR instruction to check.

TA’s advice: There’s no need to write a line (the bolded line) to make a special judgment for the first node, you can merge it with loop.

```
Iterate      LDI      R1,Head
             BRz      Exit              ;check if the linked list is empty
```

2.How to handle null input

I do nothing special, just use my compare algorithm as usual. All comparisons finish at the first loop. Just check weather the first char is ‘x0000’.