ICS LAB2 REPORT

Main Algorithm

Using blkw .16 to open a storeage area to store what the user input. Using trap x20 and trap x21 to get and ouput the character. When user ending input, shift r1 to the head of the string. When do the traversal, use r2 to store the node pointer, use r4 to denote whether there exist a person's first or last name is what we input. Then compare the first name, if equal, we jump to a branch, else we jump to another branch. And when the one we check is qualified, we output his fullname or halfname(based on whether his first or last name are both the same as we input), and add 1 to r4. When we reach the end of the list, we check whether the r4 is 0. If it is , then we output "Not Found". Then we halt the machine.

Essential Part

Get the user input.

```
getChar trap x20
        trap x21
        ld r7,antiEnter ;;just debug
        ;add r0,r0, #0
        add r7,r7, r0
        brz inputOver
        str r0,r1, #0
        add r1,r1, #1
        brnzp getChar
inputOver
        and r7,r7, #0
        str r7, r1, #0
        lea r1,strInput ;; 输入结束，输入的字符串存储在strInput中，r1指向输入的字符串
```

Check whether the person's name is what we want.

```
check   brz done
      ldr r0,r2, #2 ;; 指向first name
        JSR strcmp
        add r5,r5, #0
        brz fNo ;表示第一个不匹配
fYes
        ldr r0,r2, #3
        JSR strcmp
        add r5, r5, #0
        brz echoFull
        ;; 只输出一次名字
        brnzp echoHalf
fNo
        ldr r0,r2, #3
        JSR strcmp
        add r5, r5, #0
        brz caseDone ;;失败
        ;; 只有第二个符合，输出结果（全名+房间）
```

Output the result.

```
echoFull
        ldr r0,r2, #2
        trap x22
        ld r0, space
        trap x21
echoHalf
        ldr r0,r2, #3
        trap x22
        ld r0, space
        trap x21
        ;;输出房间号
        ldr r0,r2, #1
        trap x22
        ld r0, newLine
        trap x21 ;;换行
        add r4,r4, #1 ;;将r4加一，表示找到了
caseDone ldr r2,r2, #0
        brnzp check
done    add r4,r4, #0
        brnp finish
notFound
        lea r0,notFoundOutput
        trap x22
finish    halt
```

Compare two string saved in r0 and r1.

```
;; 以下代码来自书本第302页
;; 比较存储在r0和r1指向的地址的两个字符串，最终如果r5是0，那么不相等，否则相等
strcmp  st r0,comSaver0
        st r1,comSaver1
        st r2,comSaver2
        st r3,comSaver3
        st r4,comSaver4

        and r5,r5, #0
        add r5,r5, #1

        ldr r2,r0, #0
      brz checkr1 ;判定是否是空串

nextChar ldr r3,r1 #0
      not r3,r3
      add r3,r3, #1
      add r3, r2, r3
      brnp fail
      add r0,r0, #1
      add r1,r1, #1
      ldr r2,r0, #0
      brnp nextChar
checkr1 ldr r3,r1, #0
      brz comDone
fail  and r5,r5, #0
      brnzp comDone

comDone    ld r0,comSaver0
        ld r1,comSaver1
        ld r2,comSaver2
        ld r3,comSaver3
        ld r4,comSaver4
        ret
comSaver0  .blkw 1
comSaver1  .blkw 1
comSaver2  .blkw 1
comSaver3  .blkw 1
comSaver4  .blkw 1
upperbound .fill xffe0
        .end
```

Questions

1. Describe how you get the user input.

   Answer: We open a area to store what user input in the beginning of the program. Then using a loop to get input(trap x20), and immediately output it on the console(trap x21),then store to the area. When the user input 'enter', we ending the loop.

2. How you do the check.

   Answer: Firstly we load the address of the first node. Check the first and second name of it, based on the result we go to different output label (echoFull, echoHalf). Then we go to the next node. Finally check the value on r4 to decide whether we output 'Not Found', or not.