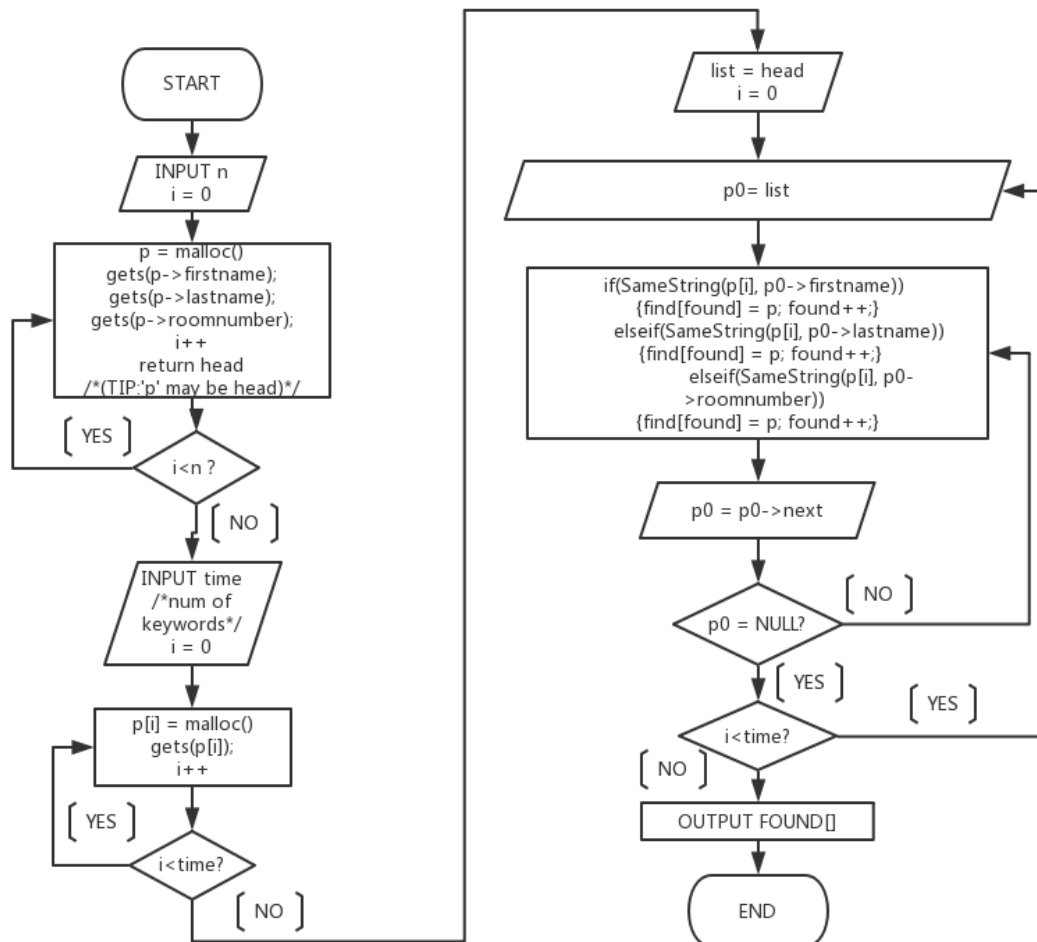


The 5th lab report

(Due on Jul.19th)

1 .Program algorithm:



2.Brief Explanations

This program is to input the information of a directory into a linked-list. Information is stored in the nodes of linked-list, each of which contains three members: first name, last name, room number and a struct pointer that points to next node.

1) FUNCTION1: dir *CreateList (int n)

This function is to create a linked-list, take over afferent information and store them into the linked-list. If the head of linked-list is NULL (which means we haven't input any information yet), allocate area for it , put the received information into it , point "end" pointer to it and point the "head->next" pointer to NULL; If the head

isn't NULL, allocate area for end->next, point end to it and input the received information into it until all the information has been stored. At that time end->next will be NULL.

To distinguish blank and '\0', we use gets() to input strings.

2) **FUNCTION2:** void SearchList (char a[], int n, dir *list)

This function is to search for the corresponding node and print its member on the screen. We use a pointer of list to go through all the nodes and comparing their elements with the given ones with strcmp(). If so, "found" plus 1 and store the pointer value into an array. At last print the number and the element on the screen.

3) **MAIN**

In MAIN, we should input the number of node, the number of nodes you want to search for and the keywords. To fit in with the output example, we firstly store these information in an array and then use it.

3.Source Code (in appendix)