Assignment Report 5

1. Solution

First, create the linked list and read the data. The reading procedure is not simple "scanf" or "gets". I wrote a function to read, which can prevent the situation when the names have SPACE.

```
/************************************************
 Function: // GetStr
 Description: // Use a while loop to read the data
 (Assure that SPACE will be read and ENTER won't)
 Calls: // none
 Called By: // main
 Input: //  From keyboard
 Output: // none
 Return: // none
 Others: // none
 ************************************************/
void GetStr(char *str)
{
    char c;
    char s[MAXNUM];
    int len = 0;
    while ((c = getchar()) == '\n');
    do {
        s[len++] = c;
    } while ((c = getchar()) != '\n');
    s[len] = '\0';
    strcpy(str, s);
}
```

Then, for every data that require search, the procedure will run two loops. One is to count the match numbers, the other is to write all the data.

3. Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAXNUM 20

struct node
{
   char First[MAXNUM];
   char Last[MAXNUM];
```

```
    char Room[MAXNUM];
    struct node *next;
}; /*Linked List*/

struct node *p, *head, *tail;
char temp[MAXNUM];

/************************************************
 Function: // GetStr
 Description: // Use a while loop to read the data
 (Assure that SPACE will be read and ENTER won't)
 Calls: // none
 Called By: // main
 Input: //  From keyboard
 Output: // none
 Return: // none
 Others: // none
 ************************************************/
void GetStr(char *str)
{
    char c;
    char s[MAXNUM];
    int len = 0;
    while ((c = getchar()) == '\n'); /*If it is not ENTER*/
    do {
        s[len++] = c;
    } while ((c = getchar()) != '\n');
    s[len] = '\0';
    strcpy(str, s);
}

/************************************************
 Function: // Input
 Description: // Create and read the data
 Calls: // GetStr
 Called By: // main
 Input: //  From keyboard
 Output: // none
 Return: // none
 Others: // none
 ************************************************/
void Input()
{
    int n, i;
```

```
    int size = sizeof(struct node);
    head = tail = NULL;
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        p = (struct node *)malloc(size);
        GetStr(p ->First);
        GetStr(p ->Last);
        GetStr(p ->Room);
        p ->next = NULL;
        if (head == NULL)
            head = p;
        else
            tail ->next = p;
        tail = p;
    }
}

/************************************************
 Function: // Search and write the result
 Description: // For every data that require search,
 the procedure will run two loops. One is to count the
 match numbers, the other is to write all the data
 Calls: // GetStr
 Called By: // main
 Input: //  none
 Output: // The required data
 Return: // none
 Others: // none
 ************************************************/
void SearchAndOutput()
{
    int m, i, flag;
    scanf("%d", &m);
    for (i = 0; i < m; i++) {
        GetStr(temp);
        flag = 0;
        p = head;
        while (p != NULL) {
            if ((strcmp(temp, p ->Last) == 0) || (strcmp(temp, p ->First) == 0)) {
                flag++;
            }
            p = p ->next;
        }
        printf("%d", flag);
```

```
        if ((flag != 0) || (i != m-1 && flag == 0)) printf("\n");
        /*Make sure the last line does not follow an ENTER*/
        p = head;
        if (flag != 0) {
            while (p != NULL) {
                if ((strcmp(temp, p ->Last) == 0) || (strcmp(temp, p ->First) == 0)) {
                    printf("%s %s %s", p ->Last, p ->First, p ->Room);
                    flag--;
                    if ((flag != 0) || (i != m-1 && flag == 0)) printf("\n");
                    /*Make sure the last line does not follow an ENTER*/
                }
                p = p ->next;
            }
        }
    }
}

int main(int argc, const char * argv[]) {
    Input();
    SearchAndOutput();
    return 0;
}
```

4. Snapshots (From Xcode)

```
4
Xiaohong
Jiang
520
Jiang
Jiang
233
Wenzhi
Chen
203
Chun
Chen
100
3
Jiang
Chu
Chun
2
Jiang Xiaohong 520
Jiang Jiang 233
0
1
Chen Chun 100Program ended with exit code: 0
```