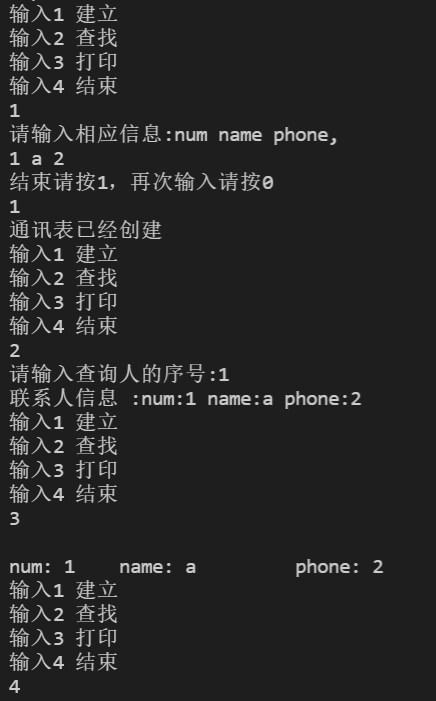
**测试用例**

建立 num = 1，name = a，phone = 2



**算法和数据结构**

1. 建立哈希函数。对输入的数据，利用公式k=10\*k+num[i]-48;和k = (k % 19);使得输入的数据和某一值有对应关系
2. 建立通讯录，对于输入的信息，利用链表存储，将输入的序号用哈希函数处理，返回k值
3. 搜索通讯录，根据需要查询的人的编号，输入哈希函数中，得到实际序号，根据这一实际序号，在通讯录中依次查找
4. 打印，从第一个序号开始，依次打印联系人的所有信息

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

typedef struct node{

    char num[11], name[15], phone[20];

    struct node \*next;

} Num;

Num \*num\_list[19];

int hash(char num[]);

void Create();

void search();

void list();

int hash(char num[]){

    int i, k = 0;

    for(i = 0; num[i] != '\0'; i++)

        k=10\*k+num[i]-48;

    k = (k % 19);

    return k;

}

void Create(){

    struct node \*p1;

    int k, m = 0;

    while(m == 0){

        printf("请输入相应信息:num name phone,\n");

        p1 = (struct node \*)malloc(sizeof(struct node));

        scanf("%s%s%s", p1->num, p1->name, p1->phone);

        k = hash(p1->num);

        p1->next = num\_list[k];

        num\_list[k] = p1;

        printf("结束请按1，再次输入请按0\n");

        scanf("%d", &m);

    }

    printf("通讯表已经创建\n");

}

void search(){

    char num[11];

    int k;

    struct node \*f;

    printf("请输入查询人的序号:");

    scanf("%s", num);

    k = hash(num);

    f = num\_list[k];

    while(f != NULL){

        if(strcmp(f->num, num) == 0)

            printf("所要查找的联系人信息 :num:%s name:%s phone:%s\n",f->num,f->name,f->phone);

        f = f->next;

    }

}

void list(){

    struct node \*f;

    int i;

    printf("打印通讯录如下：\n");

    for(i = 0; i < 19; i++){

        f = num\_list[i];

        while(f != NULL){

            printf("--->num: %s\t  name: %s\t  phone: %s\t\n", f->num, f->name, f->phone);

            f = f->next;

        }

    }

}

int main() {

    int i;

    char x;

    for(i = 0; i < 19; i++)

        num\_list[i]=NULL;

    while(1) {

        printf("输入1 建立\n");

        printf("输入2 查找\n");

        printf("输入3 打印\n");

        printf("输入4 结束\n");

        scanf("%s", &x);

        switch(x) {

        case '1':

            Create();

            break;

        case '2':

            search();

            break;

        case '3':

            list();

            break;

        case '4':

            return 0;

        default:

            printf("请重新输入；\n");

        }

    }

}