**数据结构上机实验题报告**

题目：哈希表设计

姓名：王小龙 班级：2020211310 学号：2020211502 提交日期:2021.12.12

一 题目

针对某个集体中的“人名”设计一个哈希表，使得平均查找长度不超过R，完成该哈希表的创建和查找程序;

人名为中国人姓名的汉语拼音形式，待填入哈希表的人名共有30个；

要求平均查找长度的上限为2；

哈希函数以除留余数法构造，用伪随机探测再散列法处理冲突；

测试数据为选用开发者的30个同学的姓名；

在输入人名过程中能自动识别非法输入，并给与非法输入的反馈信息要求重新输入；

查找成功时，显示姓名及关键字，并计算和输出查找成功的平均查找长度。

二 程序设计

数据类型定义：

typedef struct NAME

{

const char \*py; //名字的拼音

int k; //拼音所对应的整数

}NAME;

typedef struct hterm //哈希表

{

const char \*py; //名字的拼音

int k; //拼音所对应的整数

int si; //查找长度

}HASH;

基本操作：

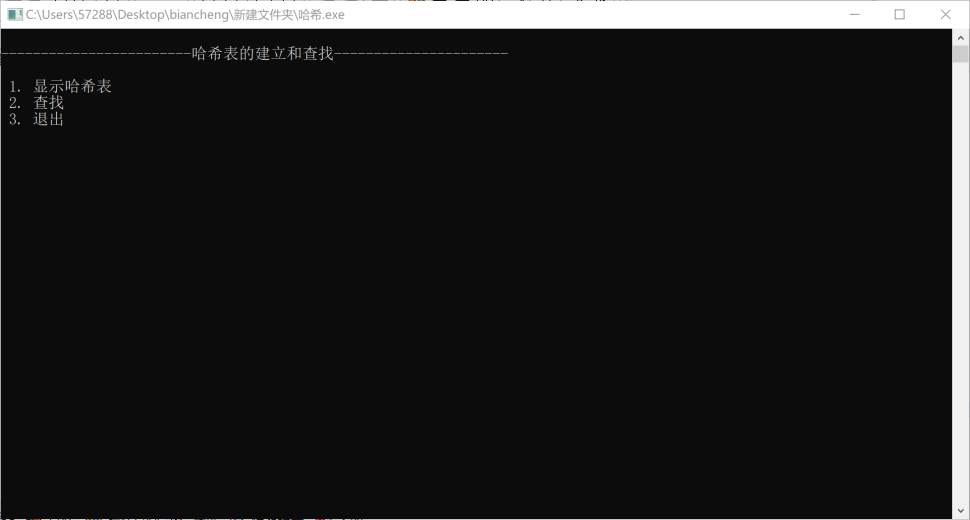
初始化函数：void InitNameList()//读取姓名信息，初始化成姓名表。

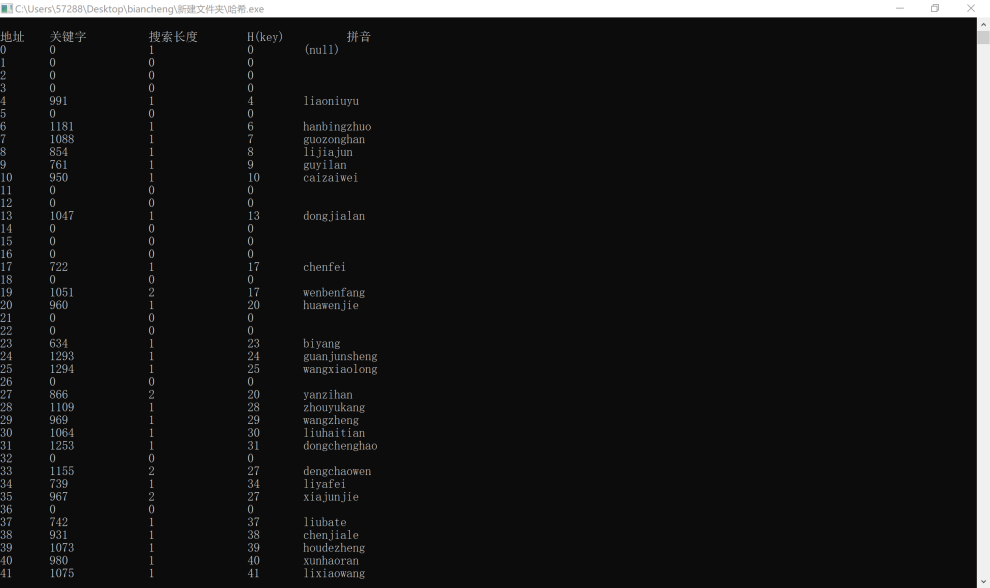
创建哈希表函数：void CreateHashList()//将读取信息建立的姓名表建立哈希表。

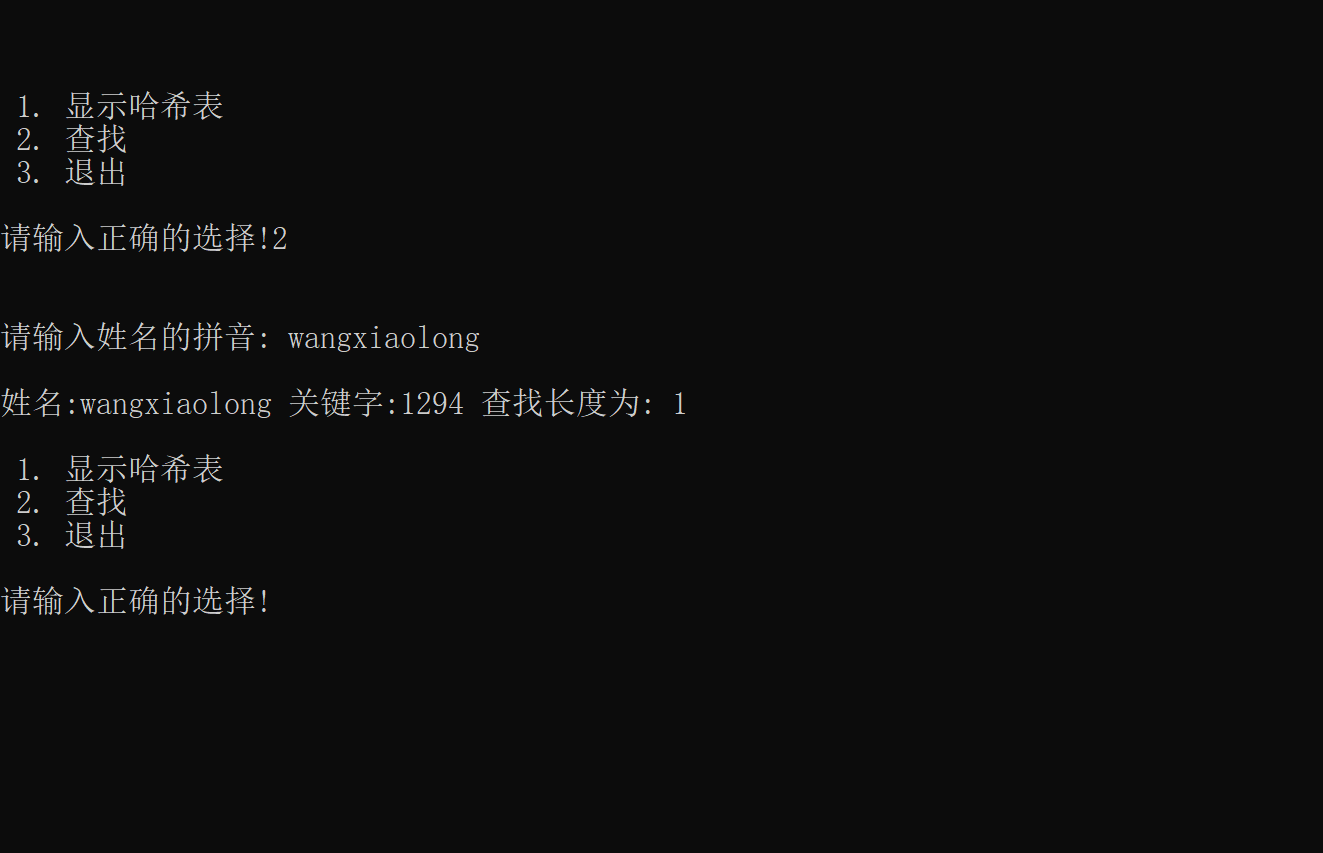
显示哈希表函数：void Display()//打印哈希表中的信息。

查找哈希表中的姓名函数 void FindList() //按照输入的姓名查找其在哈希表中的位置，并打印信息。

三 程序测试运行







四 编程工作总结

要注意正确的人名信息输入，不要超过了限制。

五 程序源代码

#include <stdio.h>

#include<malloc.h>

#include<string.h>

#define HASH\_LEN 50 //哈希表的长度

#define M 47

#define NAME\_NO 30 //人名的个数

typedef struct NAME

{

const char \*py; //名字的拼音

int k; //拼音所对应的整数

}NAME;

NAME NameList[HASH\_LEN];

typedef struct hterm //哈希表

{

const char \*py; //名字的拼音

int k; //拼音所对应的整数

int si; //查找长度

}HASH;

HASH HashList[HASH\_LEN];

/\*-----------------------姓名（结构体数组）初始化---------------------------------\*/

void InitNameList()

{

NameList[0].py="huawenjie";

NameList[1].py="guyilan";

NameList[2].py="guozonghan";

NameList[3].py="hanbingzhuo";

NameList[4].py="liyafei";

NameList[5].py="liaoniuyu";

NameList[6].py="yanzihan";

NameList[7].py="zhouyukang";

NameList[8].py="biyang";

NameList[9].py="chenfei";

NameList[10].py="chenjiale";

NameList[11].py="caizaiwei";

NameList[12].py="dongchenghao";

NameList[13].py="dengchaowen";

NameList[14].py="dongjialan";

NameList[15].py="guanjunsheng";

NameList[16].py="houdezheng";

NameList[17].py="liubate";

NameList[18].py="liuhaitian";

NameList[19].py="lijiajun";

NameList[20].py="lixiaowang";

NameList[21].py="wenbenfang";

NameList[22].py="wangxiaolong";

NameList[23].py="wangyubo";

NameList[24].py="wangzheng";

NameList[25].py="xunhaoran";

NameList[26].py="xuhaoyang";

NameList[27].py="xiajunjie";

NameList[28].py="yangtong";

NameList[29].py="yuze";

const char \*f;

int r,s0;

for (int i=0;i<NAME\_NO;i++)

{

s0=0;

f=NameList[i].py;

for (r=0;\*(f+r) != '\0';r++)

s0=\*(f+r)+s0;

NameList[i].k=s0;

}

}

/\*-----------------------建立哈希表---------------------------------\*/

void CreateHashList()

{

for (int i=0; i<HASH\_LEN;i++)//哈希表的初始化

{

HashList[i].py="";

HashList[i].k=0;

HashList[i].si=0;

}

int i=0;

for (i=0;i<=NAME\_NO;)

{

int sum=0;

int adr=(NameList[i].k) % M; //哈希函数

int d=adr;

if(HashList[adr].si==0) //如果不冲突

{

HashList[adr].k=NameList[i].k;

HashList[adr].py=NameList[i].py;

HashList[adr].si=1;

}

else //冲突

{

do

{

d=(d+((NameList[i].k))%10+1)%M; //伪散列

sum=sum+1; //查找次数加1

}while (HashList[d].k!=0);

HashList[d].k=NameList[i].k;

HashList[d].py=NameList[i].py;

HashList[d].si=sum+1;

}i++;

}

}

/\*-------------------------------------查找------------------------------------\*/

void FindList()

{

printf("\n\n请输入姓名的拼音: "); //输入姓名

char name[20]={0};

scanf("%s",name);

int s0=0;

for (int r=0;r<20;r++) //求出姓名的拼音所对应的整数(关键字)

s0+=name[r];

int sum=1;

int adr=s0 % M; //使用哈希函数

int d=adr;

if(HashList[adr].k==s0) //分3种情况进行判断

printf("\n姓名:%s 关键字:%d 查找长度为: 1",HashList[d].py,s0);

else if (HashList[adr].k==0)

printf("无该记录!");

else

{

int g=0;

do

{

d=(d+s0%10+1)%M; //伪散列

sum=sum+1;

if (HashList[d].k==0)

{

printf("无记录! ");

g=1;

}

if (HashList[d].k==s0)

{

printf("\n姓名:%s 关键字:%d 查找长度为:%d",HashList[d].py,s0,sum);

g=1;

}

}while(g==0);

}

}

/\*--------------------------------显示哈希表----------------------------\*/

void Display()

{

printf("\n\n地址\t关键字\t\t搜索长度\tH(key)\t\t拼音 \n"); //显示的格式

for(int i=0; i<15; i++)

{

printf("%d ",i);

printf("\t%d ",HashList[i].k);

printf("\t\t%d ",HashList[i].si);

printf("\t\t%d ",(HashList[i].k)%M);

printf("\t %s ",HashList[i].py);

printf("\n");

}

// printf("按任意键继续显示...\n");

// getch();

int i=15;

for( i=15; i<30; i++)

{

printf("%d ",i);

printf("\t%d ",HashList[i].k);

printf("\t\t%d ",HashList[i].si);

printf("\t\t%d ",(HashList[i].k)%M);

printf("\t %s ",HashList[i].py);

printf("\n");

}

// printf("按任意键继续显示...\n");

// getch();

for( i=30; i<40; i++)

{

printf("%d ",i);

printf("\t%d ",HashList[i].k);

printf("\t\t%d ",HashList[i].si);

printf("\t\t%d ",(HashList[i].k)%M);

printf("\t %s ",HashList[i].py);

printf("\n");

}

//printf("按任意键继续显示...\n");

//getch();

for( i=40; i<50; i++)

{

printf("%d ",i);

printf("\t%d ",HashList[i].k);

printf("\t\t%d ",HashList[i].si);

printf("\t\t%d ",(HashList[i].k)%M);

printf("\t %s ",HashList[i].py);

printf("\n");

}

float average=0;

for (i=0;i<HASH\_LEN;i++)

average+=HashList[i].si;

average/=NAME\_NO;

printf("\n\n平均查找长度：ASL(%d)=%f \n\n",NAME\_NO,average);

}

/\*--------------------------------主函数----------------------------\*/

int main()

{

printf("\n------------------------哈希表的建立和查找----------------------");

InitNameList();

CreateHashList ();

while(1)

{

printf("\n\n");

printf(" 1. 显示哈希表\n");

printf(" 2. 查找\n");

printf(" 3. 退出\n");

err:

char ch1;

scanf("%c",&ch1);

if (ch1=='1')

Display();

else if (ch1=='2')

FindList();

else if (ch1=='3')

return 0;

else

{

printf("\n请输入正确的选择!");

goto err;

}

}

}