实验报告

## 实验名称 ：图遍历

## 需求分析

题目的意思为：

对30个人的名字进行设计一个哈希表，使得哈希表平均查找长度少于R。

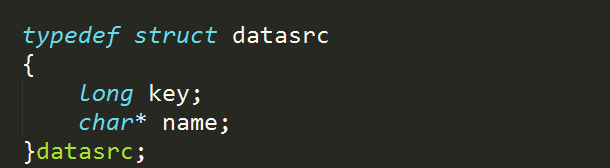
首先需要完成三个功能：

1. 读入数据
2. 哈希表建表
3. 由用户输入来查询

## 概要设计

1. 读入数据

从文件里读入并读入一个数组，用如下数组来存储:



结构中的name存储每一个人的人名，key为由name所的到的hash值，下边会用到。

1. 哈希表建表。

首先得得到每一个name的key值，然后再进行hash。

由函数transkey来完成 ：



接下由start\_hash函数来完成对name进行建立哈希表



1. 由用户输入来查询

写一个find函数来进行查找

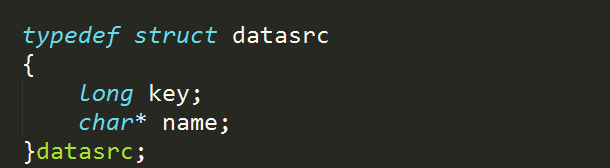


查找时需要用户输入信息来进行配合。

## 详细设计.

1.读入数据

从文件里读入并读入一个数组，用如下数组来存储:



结构中的name存储每一个人的人名，key为由name所的到的hash值，下边会用到。

用c++的文件输入流，读入in.txt中的数据，然后进行读入存入到name中的name中即可。

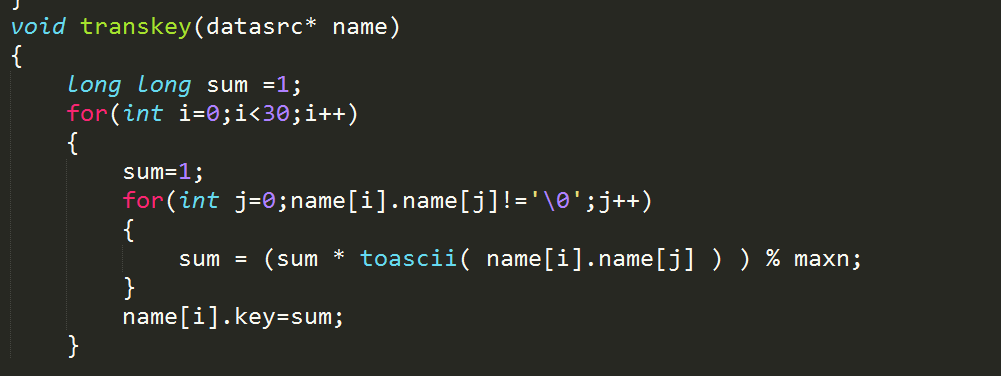
2.哈希表建表。

首先得得到每一个name的key值，然后再进行hash。

由函数transkey来完成 ：



在函数中，需要对上边的name中的那么得到key值，并存入name的key中。此时由于是字母，想要充分分散，就用每一个字母的arcii编码做乘积，然后%一个maxn，即宏定义的最大值，即可得到对应的key值。



接下由start\_hash函数来完成对name进行建立哈希表



首先按照题目的意思，是要用除留余数法来构造哈希函数，由于我定义了哈希表表长为50，故去k最大为47。用伪随机探测来处理冲突。哈希函数如下:



伪随机函数由两处组成：





由于这样看的不太清楚，可看附带的随机函数测试代码！

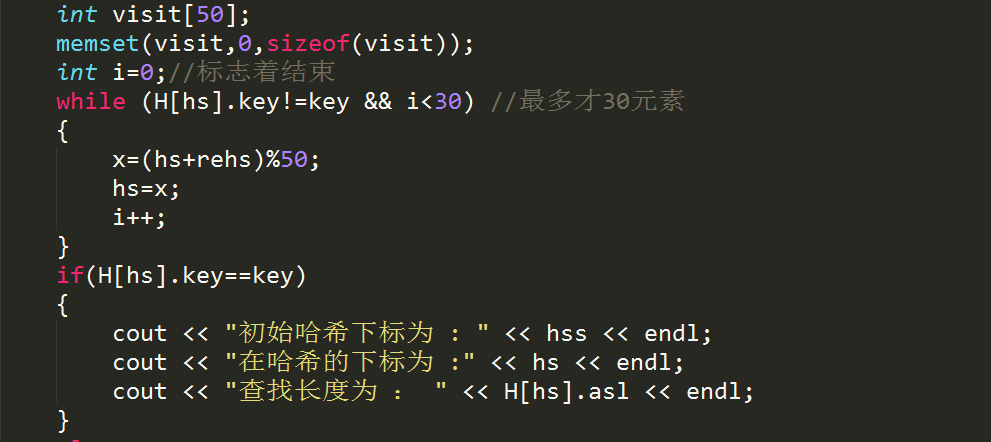
解决了这两大难题，就很easy的写出来了，内容如下：



3.find函数

首先由用户输入查询的人名，然后，得到该人名的hash值。

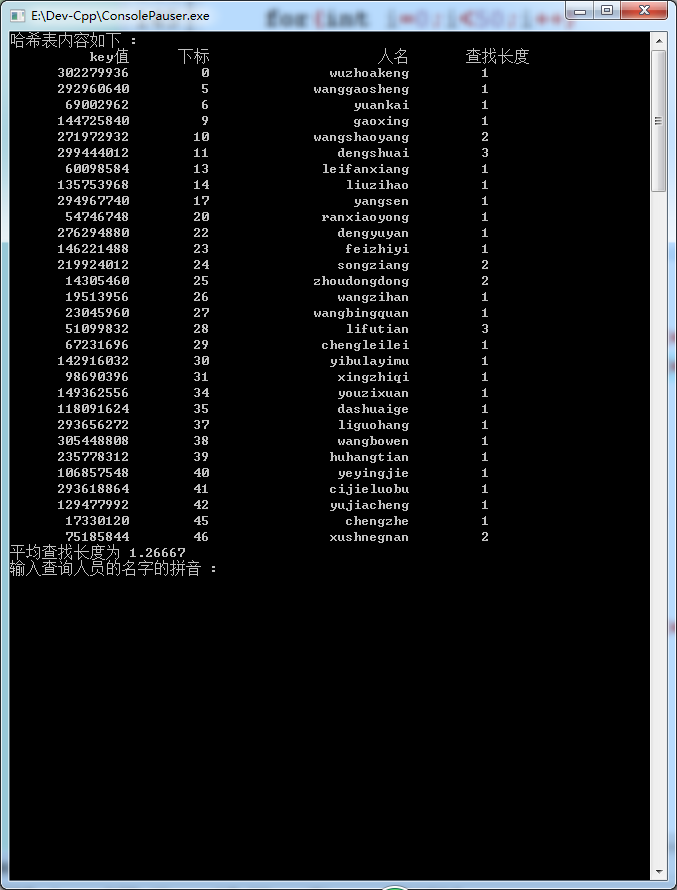
根据哈希函数的到f(key)的值。然后若出现冲突，用伪随机探测来完成的到下一位置。如下图所示：



## 用户手册

**由于输入数据全在in.txt中。此处不再赘述。**

**程序建好表之后会要求用户输入查询的人名。并且已经列出了平均查找长度。如下：**



输入上边列出来的人名进行查询即可。平均好找长度为1.26.

## 附录

输入数据文件名称: in..txt

主要程序: 哈希表的设计.cpp

附加验证程序：伪随机函数的检验.cpp