Бюджетное учреждение высшего образования Ханты-Мансийского автономного округа – Югры «СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт

Кафедра автоматики и компьютерных систем

Отчет

по лабораторной работе № 7

по дисциплине «Алгоритмы и структуры данных»

Проверила: старший преподаватель кафедры АиКС,

Карамышев. Э. Р

Выполнил: студент группы 605-31,

Хайитов Ш. Д.

Сургут 2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc194868567)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc194868568)

[ЗАКЛЮЧЕНИЕ 10](#_Toc194868569)

# **ВВЕДЕНИЕ**

**Цель:** изучить принципы построения хеш-функции, обладающих равномерным распределением, исследовать статические свойства хеш-функций, закрепить навыки структурного программирования.

**Задачи:**

1. Разработать и реализовать функцию, осуществляющую хеширование данных(Тип данных определяется вариантом).
2. Разработать и реализовать функцию-генератор, осуществляющую формирование значений ключей. Генерируемые ключи должны быть уникальными.
3. Исследовать статистические свойства разработанной хеш-функции при заданных размерах хеш-таблицы и количество ключей.
4. Составить отчет, в котором привести листинг хеш-функции, гистограммы распределения индексов, формируемых хеш-функцией(для двух значений размера хеш-таблицы) и вывод по работе(дать оценку зависимости от размера таблицы и от природы исходных данных - если таковые имеются; оценить качество разработанной хеш-функции).

Вариант данный преподавателем представлен на рисунке 1.

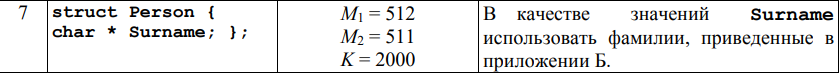


Рисунок 1 – Варинат 7

# **ОСНОВНАЯ ЧАСТЬ**

Листинг функции проверки фамилии:

int Exists **(**person**\*** perArr**,** person**\*** per**)** **{**

**for** **(**int i **=** 0**;** i **<** K**;** i**++){**

**if** **((**strcmp**(**perArr**[**i**].**Surname**,** per**->**Surname**)** **==** 0**)** **&&** **(**strcmp**(**perArr**[**i**].**Surname**,** "none"**)** **!=** 0**))** **{**

**return** 1**;**

**}**

**return** 0**;**

**}**

**}**

Листинг функции для прочитки фамилии:

int ReadPerson**(**person**\*** perArr**)** **{**

FILE**\*** inputFile **=** fopen**(**"surname.txt"**,** "r"**);**

**if** **(!**inputFile**)** **{**

**return** 0**;**

**}**

char buff**[**SIZE**];**

int i **=** 0**;**

**while** **(**fscanf**(**inputFile**,** "%s"**,** buff**)** **==** 1**)** **{**

person t**;**

strcpy**(**t**.**Surname**,** buff**);**

**if** **(**Exists**(**perArr**,** **&**t**))** **{**

sprintf**(**perArr**[**i**].**Surname**,** "%s%d"**,** buff**,** i**);**

**}else** **{**

strcpy**(**perArr**[**i**].**Surname**,** buff**);**

**}**

i**++;**

**}**

fclose**(**inputFile**);**

**return** 1**;**

**}**

Определение хэша:

uint32\_t GetHash**(**const person**\*** key**)** **{**

uint32\_t hash **=** 0**;**

**for** **(**int i **=** 0**;** i **<** strlen**(**key**->**Surname**);** i**++){**

hash **=** **(**hash **<<** 2**)** **-** hash **+** key**->**Surname**[**i**];**

**}**

**return** hash **%** M**;**

**}**

Гистограмма для M = 511 представлено на рисунке 2.

A graph of blue lines

AI-generated content may be incorrect.

Рисунок 2 – M = 511

Гистограмма для M = 512 представлено на рисунке 3.

A graph with blue lines

AI-generated content may be incorrect.

Рисунок 3 – M = 512

**Листинг программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <stdint.h>

#include <math.h>

#define M 513

#define K 2000

#define SIZE 128

**typedef** struct Person **{**

char Surname**[**SIZE**];**

**}**person**;**

int Exists **(**person**\*** perArr**,** person**\*** per**)** **{**

**for** **(**int i **=** 0**;** i **<** K**;** i**++){**

**if** **((**strcmp**(**perArr**[**i**].**Surname**,** per**->**Surname**)** **==** 0**)** **&&** **(**strcmp**(**perArr**[**i**].**Surname**,** "none"**)** **!=** 0**))** **{**

**return** 1**;**

**}**

**return** 0**;**

**}**

**}**

int ReadPerson**(**person**\*** perArr**)** **{**

FILE**\*** inputFile **=** fopen**(**"surname.txt"**,** "r"**);**

**if** **(!**inputFile**)** **{**

**return** 0**;**

**}**

char buff**[**SIZE**];**

int i **=** 0**;**

**while** **(**fscanf**(**inputFile**,** "%s"**,** buff**)** **==** 1**)** **{**

person t**;**

strcpy**(**t**.**Surname**,** buff**);**

**if** **(**Exists**(**perArr**,** **&**t**))** **{**

sprintf**(**perArr**[**i**].**Surname**,** "%s%d"**,** buff**,** i**);**

**}else** **{**

strcpy**(**perArr**[**i**].**Surname**,** buff**);**

**}**

i**++;**

**}**

fclose**(**inputFile**);**

**return** 1**;**

**}**

uint32\_t GetHash**(**const person**\*** key**)** **{**

uint32\_t hash **=** 0**;**

**for** **(**int i **=** 0**;** i **<** strlen**(**key**->**Surname**);** i**++){**

hash **=** **(**hash **<<** 2**)** **-** hash **+** key**->**Surname**[**i**];**

**}**

**return** hash **%** M**;**

**}**

void PrintPerSur**(**person**\*** perArr**)** **{**

**for** **(**int i **=** 0**;** i **<** K**;** i**++)** **{**

printf**(**"0x%03x %s\n"**,** GetHash**(&**perArr**[**i**]),** perArr**[**i**].**Surname**);**

**}**

**}**

int main**()** **{**

person personArr**[**K**];**

**for** **(**int i **=** 0**;** i **<** K**;** strcpy**(**personArr**[**i**].**Surname**,** "none"**),** i**++);**

**if** **(!**ReadPerson**(**personArr**))** printf**(**"[ERR] Open a file"**);**

int personHashMap**[**M**]** **=** **{** 0 **};**

**for** **(**int i **=** 0**;** i **<** K**;** i**++)** **{**

personHashMap**[**GetHash**(&**personArr**[**i**])]++;**

**}**

**for** **(**int i **=** 0**;** i **<** M**;** i**++)** **{**

printf**(**"%d %d\n"**,** i **+** 1**,** personHashMap**[**i**]);**

**}**

double k **=** M **/** **(**double**)**K**;**

double k\_ **=** K **/** **(**double**)**M**;**

double stats **=** 0**;**

**for** **(**int i **=** 0**;** i **<** M**;** i**++)** **{**

stats **+=** **(**personHashMap**[**i**]** **-** k\_**)** **\*** **(**personHashMap**[**i**]** **-** k\_**);**

**}**

printf**(**"M=%d N=%d\n"**,** M**,** K**);**

printf**(**

"%lf [%lf %lf]"**,**

k **\*** stats**,**

M **-** sqrt**(**M**),**

M **+** sqrt**(**M**)**

**);**

**return** 0**;**

**}**

# **ЗАКЛЮЧЕНИЕ**

В ходе лабораторной работы была реализована хеш-функция. Были изучены принципы построения хеш-функций, так же были исследованы их статистические свойства. При увеличении размера таблицы, коэффициент хи-квадрат увеличивается.