БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

Отчет к лабораторной работе №6 на тему

**«Моделирование хеш-таблиц»**

в рамках курса «Аппаратные основы интеллектуальных систем»

Выполнила: Рабушка А.А., 021703

Проверил: Захаров В.В.

Минск, 2021

**Цель работы**

Освоение навыков построения и проверки таблиц хеширования.

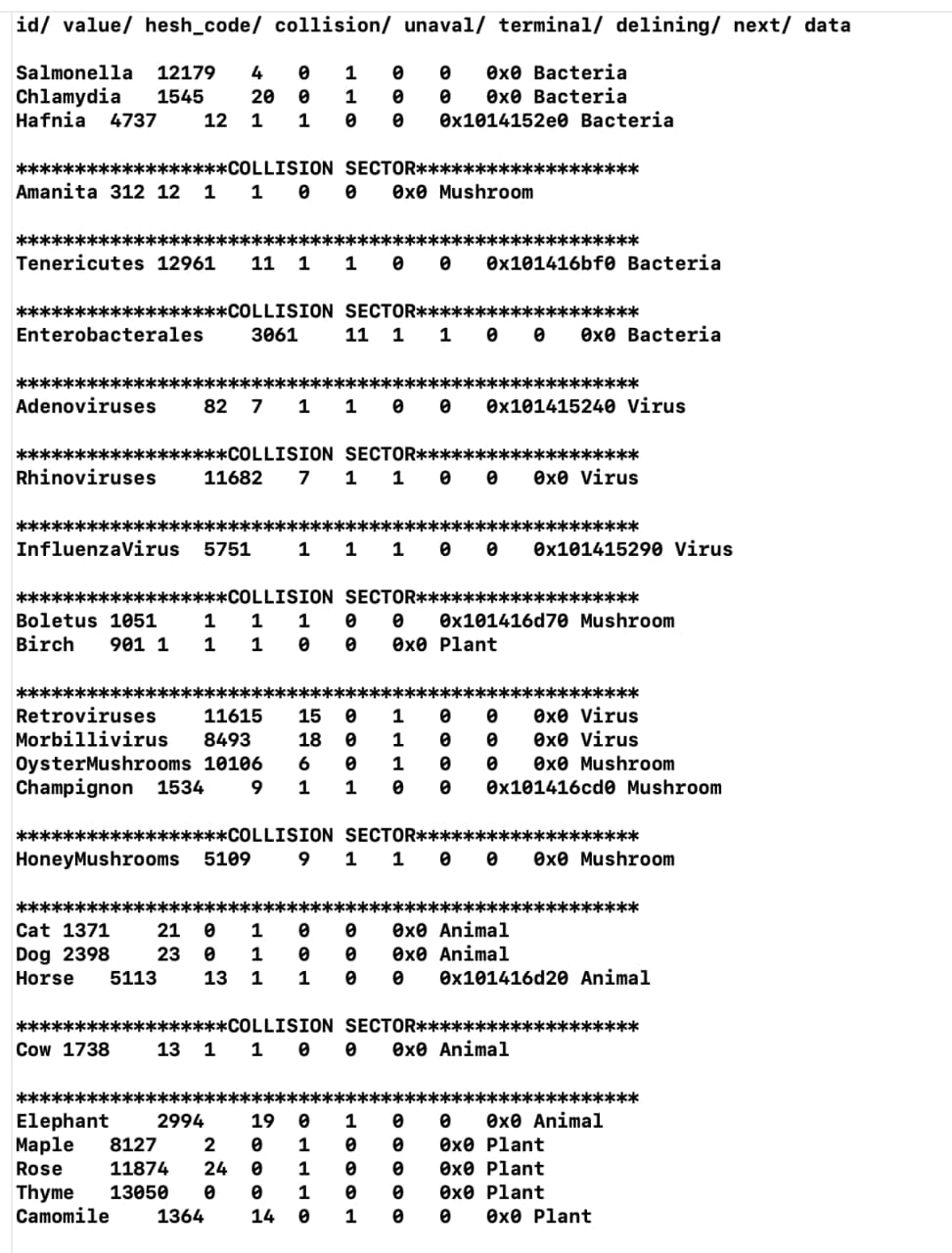
Задания

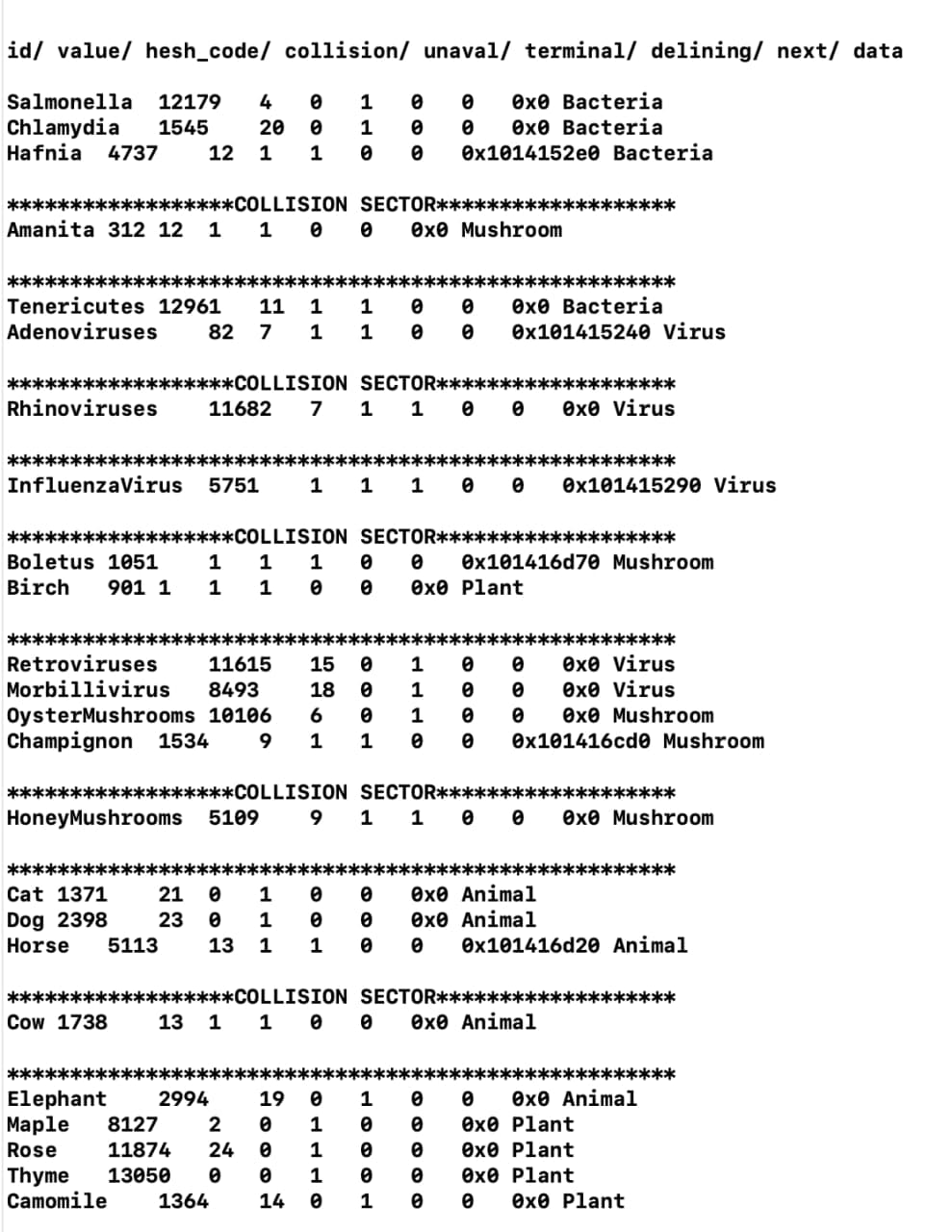
1. Разработать и проверить программу, обеспечивающую формирование хеш-таблицы, по ключевым словам и выполнение различных операций с этой таблицей – включение в таблицу новых строк, поиск информации в таблице по ключевым словам, удаление строк из таблицы.
2. Хеш-таблица должна быть тематической, т.е. все ключевые слова и данные должны быть из определенной отрасли науки, техники или деятельности людей. В работе реализован вариант 14, соответствующий номеру в списке группы; тема - «Биология».
3. При выполнение любого варианта ЛР №6 должны быть выполнены следующие требования:  
    - размер таблицы не менее 20 строк;  
    - количество записанных в таблице строк – не менее 10-ти;  
    - количество коллизий не менее 2-х;  
    - количество строк, объединенных в цепочки резервных ячеек, сформированных из-за коллизий – не менее 3-х;  
    - размер и структура каждой строки хеш-таблицы должны быть одинаковыми и постоянными.

**Ход работы**

1. Создается таблица на 25 строк. Каждая строка таблицы содержит: id (ключевое слово), value (сумма номеров первых n символов), hesh\_code (хеш-код), collision (флажок коллизии), unaval (флажок «занято»), terminal (терминальный флажок), delining (флажок вычеркивания), next (указатель на следующий элемент в области коллизии), data (данные).
2. В таблицу заносится 25 понятий из научной области «Биология»: каждому биологическому виду ставится в соответствие царство, к которому он относится.
3. В дальнейшем пользователь может добавлять записи и удалять их, а также очистить таблицу для создания новой. В программе был реализован соответствующий функционал:
   1. Функция, которая возвращает порядковый номер в алфавите символа, переданного ей: *alpha()*
   2. Функция, которая возвращает целочисленное число value – сумма 3 чисел, соответствующих номерам первых трех символов строки-ключа в алфавите. Если в строке меньше 3 чисел, берутся значения всех символов строки: *get\_value()*
   3. Функция, которая возвращает числовое значение val, вычисленное по формуле val = V % 10 + B, где V – значение, вычисленное функцией get\_value(), В – размер хеш-таблицы. Значение val используется в качестве хеш-ключа: *get\_hesh()*
   4. Функция, осуществляющая вставку в таблицу пары элемент-значение. По результатам работы функции get\_hesh() проводится анализ наличия коллизии: если коллизии не наблюдается – пара вставляется в следующую ячейку хеш-таблицы, иначе находится последний элемент с таким же хещ-ключом и данная пара заносится в ячейку next последнего элемента с таким же хеш-ключом. Все соответствующие данной паре флажки и значения также заносятся в таблицу согласно наличию/отсутствию коллизии: *add\_el()*
   5. Функция, осуществляющая удаление элемента из хеш-таблицы по переданному в данную функцию ключу. Отыскивается ячейка таблицы с переданным ключом и производится операция удаления данной ячейки. Если элемент находится в области коллизии, вся цепочка “разрывается” по данному элементу, элемент удаляется из цепочки и цепочка снова склеивается. Иначе ячейка просто извлекается из таблицы: *del\_el()*
   6. Функция, выводящая на экран некоторые данные, соответствующие ключевому слову, переданному в функцию. Вычисляется хеш-код данного ключевого слова и по хеш-коду осуществляется поиск и вывод на экран соответствующих данных: *print\_search\_el()*
   7. Функция, осуществляющая вывод на экран всей хеш-таблицы: *print\_table()*

**4.** На рис.1 изображена исходная таблица, содержащая 25 элементов; на рис.2 – таблица после удаления элемента «Enterobacterales»; на рис.3 – данные (соответствующее царство) элемента «Maple».

****

****



*Рис.3 – Данные элемента «Maple»*

**Вывод:**

Преобразования вида ключ-адрес широко используются в реальных задачах  представления информации, например, в словарях и таблицах символов. В частности, формирование соответствий некоторого ключа (единицы информации) и его адреса является причиной создания так называемых хеш-таблиц, которые которые состоят из блоков ключа и адреса и предоставляют доступ к информации одного рода при помощи адресов, которые генерируются хеш-функциями. Адреса, по своей сути, это реальные физические адреса в некоторой ЭВМ.

Хеш-таблица – это структура данных, позволяющая хранить пары (ключ - значение) и выполнять операции добавления новой пары, поиска и удаления пары по ключу. Хеш таблицы используются, например, для хранения истории посещения сайтов в браузере, для хеширования и хранения данных пользователей на различных ресурсах и сайтах, в аппаратных маршрутизаторах и т.д.

Одна из главных составляющих любой хеш-таблицы - метод генерации адреса, функция, которая преобразует ключ в некоторое числовое значение. Такие функции бывают разных видов: деление/умножение, сегментация, извлечение битов. В данной лабораторной работе был применен метод деления, который является достаточно удобным для хеширования слов любых алфавитов. Удобство выражается в том, что при применении такого метода наиболее низок риск совпадения хеш-адресов у разных единиц информации.

Совпадение адресов - коллизия - это одна из ключевых проблем, которая может возникнуть при хешировании. Самый простой и действенный способ разрешения этой проблемы - создание цепочки коллизий. При этом первый элемент этой цепочки в флаге Т хранит значение 1, а все последующие - 0. Такой принцип используется в современных программных реализациях хеш-таблицу, в которых каждая строка таблицы - это список, и если в списке элемент один, то, очевидно, эта строка не участвует в коллизии по ее хеш-адресу.

Хеширование может осуществляться различными способами. Если хеш-ключи известны заранее, то можно избежать коллизий, найдя для них совершенную хеш-функцию.

Для поиска свободных резервных ячеек при внутренней адресации применяется процедура пробинга (линейного, квадратичного или случайного).