Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №6 по теме «Моделирование хеш-таблиц» по курсу «АОИС»

Выполнил: Пилипейко В.И гр.921701

Проверил: Захаров В.В

МИНСК

2020

**Цель работы**: освоение навыков построения и проверки таблиц хеширования.

**Задание**

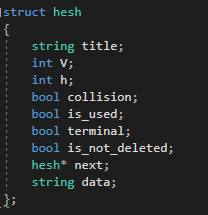
Разработать и проверить программу, обеспечивающую формирование хеш-таблицы, по ключевым словам и выполнение различных операций с этой таблицей – включение в таблицу новых строк, поиск информации в таблице  по ключевым словам, удаление строк из таблицы.

Вариант 19

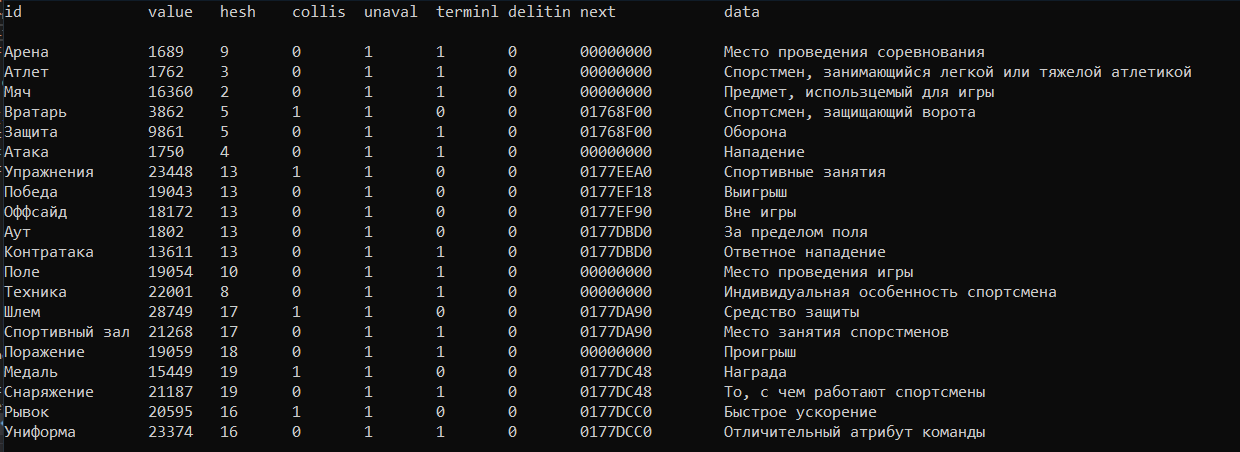
Предметная область записываемых ключевых слов - **спорт**.

**Ход и результаты работы**

1. Создадим таблицу на 20 строк. Каждая строка, как и сама таблица, будет представлять собой отдельный объект. Каждая строка состоит из полей  title и data - понятия и пояснения к нему; полей, отражающих значение флагов collision, is\_used, terminal, is\_not\_deleted  - первенство строки в цепочке коллизий, использование конкретного объекта, завершение цепочки коллизии и статус удаления соответственно - и полей, в которых записаны числа, V – числовое значение ключевого слова и h- хеш-адрес, вычисленный по числовому значению ключевого слова (h(V) = V mod H; v = [0]\*x2 + [1]\*x + [2], где x - основание системы счисления (количество букв в алфавите используемого языка).



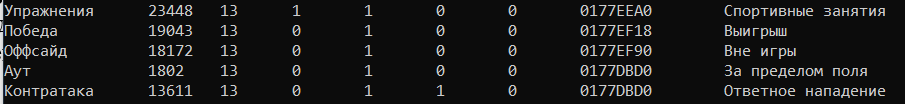
1. Занесем в таблицу 20 понятий из предметной области спорта:



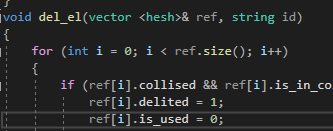
3. В дальнейшем пользователь может добавлять записи и удалять их, а также искать информацию в таблице по ключевым словам. В программе был реализован соответствующий функционал:

1) добавление записи происходит по принципу «экономии» памяти - сначала программа ищет ячейки, освободившиеся в результате удаления исходных строк, и заполняет их новыми данными, а лишь потом приступает к записи в чистые ячейки, если удаленные не были найдены (проверяется флаг deleted, если флаг равен единице, на ее место можно вставить новую запись).

2) Коллизия (совпадение у разных строк h(V)) разрешается созданием области переполнения: если h(V) первой строки совпадает с h(V) другой строки, то флаг C = 1 у первой строки, а T = 1 у той строки, которая является последней в цепочке коллизий; адрес следующей строки в цепочке коллизий будет указана в поле next данной строки, а если строка является последней в цепочке коллизий или вовсе не входит ни в какую цепочку, то next = nullptr. Флаг is\_used добавленной строки равен 1, deleted = 0.



Удаление записи: флаги строки становятся is\_used = 0, deleted = 1.

  
  
 а) если состояние флажка terminal=1 и поле next равно NULL, то эта строка является одиночной и ее можно просто удалить, сделать свободным, т.е. установить флажок is\_used в «0»

б) если состояние флажка terminal=1 и в поле next записан адрес, указывающий, что эта строка входит в цепочку резервных ячеек, то эта строка является конечной в какой-то цепочке. В этом случае отслеживается предыдущая относительно удаляемой строки в цепочке путем сравнения содержимого поля next всех предыдущих строк с адресом удаляемой строки h и далее: - удаляемая строка делается свободной (флажок U устанавливается в 0) - в найденной предыдущей строке, у которой значения поля next = адресу удаляемой строки, флажок terminal устанавливается в «1».

в) Если состояние флажка terminal=0 и содержание поля next != адресу текущей строки, то удаляемая строка является другой строкой (не последней) в цепочке резервных ячеек. В этом случае находится последующая за удаляемой строкой в этой цепочке по содержимому поля next и эта строка переписывается на место удаляемой и делается свободной, т.е. в ней устанавливается флажок is\_used в «0», фактически удаляется не вычисление по хеш-адресу строки, а следующее за ней, при этом содержимое остальных строк этой цепочки не изменяется.

г) если состояние флажка terminal=0, а collision=1, то эта строка является первой в цепочке, начинающейся с адреса удаляемой строки. В этом случае по содержимому поля next остлеживается следующая за удаляемой строка, её содержимое переписывается в удаляемую, в ней устанавливается collision=1, и эта строка делается свободной (is\_used=0).

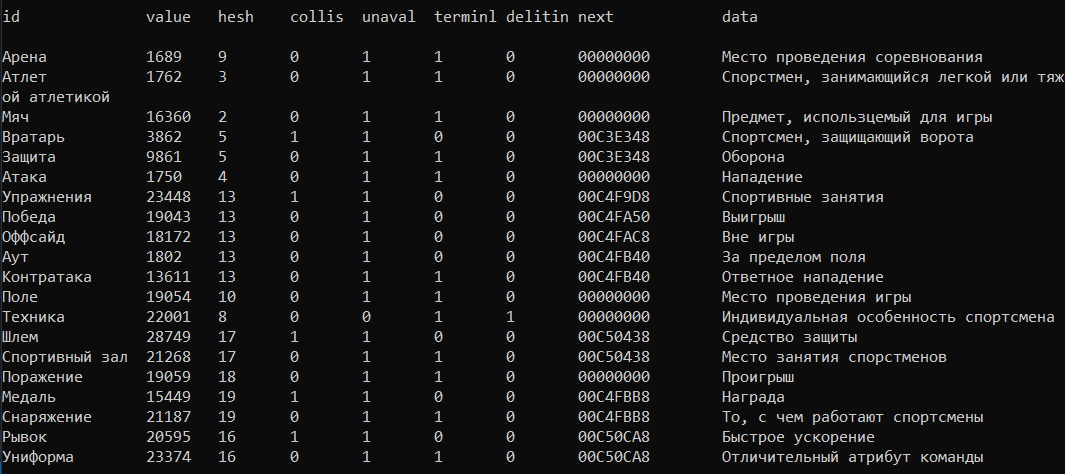


Рисунок 1. Результат удаления "техника"

3) Поиск информации в таблице по ключевым словам. Здесь по введенному ключевому слову найдется дополнительная информация о конкретном введеном слове.



Вывод

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

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

Ситуация, когда для различных ключей получается одно и то же хеш-значение, называется коллизией. В этом случае для размещения второй и последующих записей необходимо использовать резервные ячейки памяти, которые размещаются либо в самой таблице хеширования (внутренняя адресация), либо в специальной области памяти (области переполнения).

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