

## Практическая работа №3

### Алгоритм хеширования таблиц

**Цель:** Изучить алгоритмы хеширования таблиц

Хеширование (или хэширование, англ. hashing) – это преобразование входного массива данных определенного типа и произвольной длины в выходную битовую строку фиксированной длины.

Хеш-таблица – это структура данных, реализующая интерфейс ассоциативного массива, то есть она позволяет хранить пары вида "ключ-значение" и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу. Хеш-таблица является массивом, формируемым в определенном порядке хеш-функцией.

В качестве использования хеширования в повседневной жизни можно привести примеры распределение книг в библиотеке по тематическим каталогам, упорядочивание в словарях по первым буквам слов.

Доступ к элементам осуществляется по его ключу. Основные операции, которые могут выполняться с хеш-таблицей:

- Insert — добавить новый элемент в хеш-таблицу
- Delete — удалить элемент из хеш-таблицы по ключу
- Search — получить значение по ключу

Основной принцип работы хеш-таблицы заключается в том, что в качестве входных параметров она принимает пары ключ-значение. Затем с помощью специальной хеш-функции получает короткий ключ на основе полученного ключа. И, наконец, добавляет данные в таблицу. Если в таблице уже существует значения с таким хешем, то объединяет их в коллекции. Внутри коллекции поиск выполняется по изначальному полученному ключу.

#### Код

```
//Хеш – функция
unsigned int hashtable_hash(char* key) {
    unsigned int h = 0; char* p;
```

					<i>АИСД.09.03.02.050000 ПР</i>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Ермошина В.А			Практическая работа №3 Алгоритм хеширования таблиц	Лит.	Лист
Провер.		Берёза А.Н.					
Реценз						2	10
Н. Контр.						ИСОиП (филиал) ДГТУ в г.Шахты ИСТ-Тб21	
Утверд.							

```

for (p = key; *p != '\0'; p++) {
    h = h * HASHTAB_MUL + (unsigned int)*p; }
return h % HASHTAB_SIZE;
}

```

Диаграмма деятельности

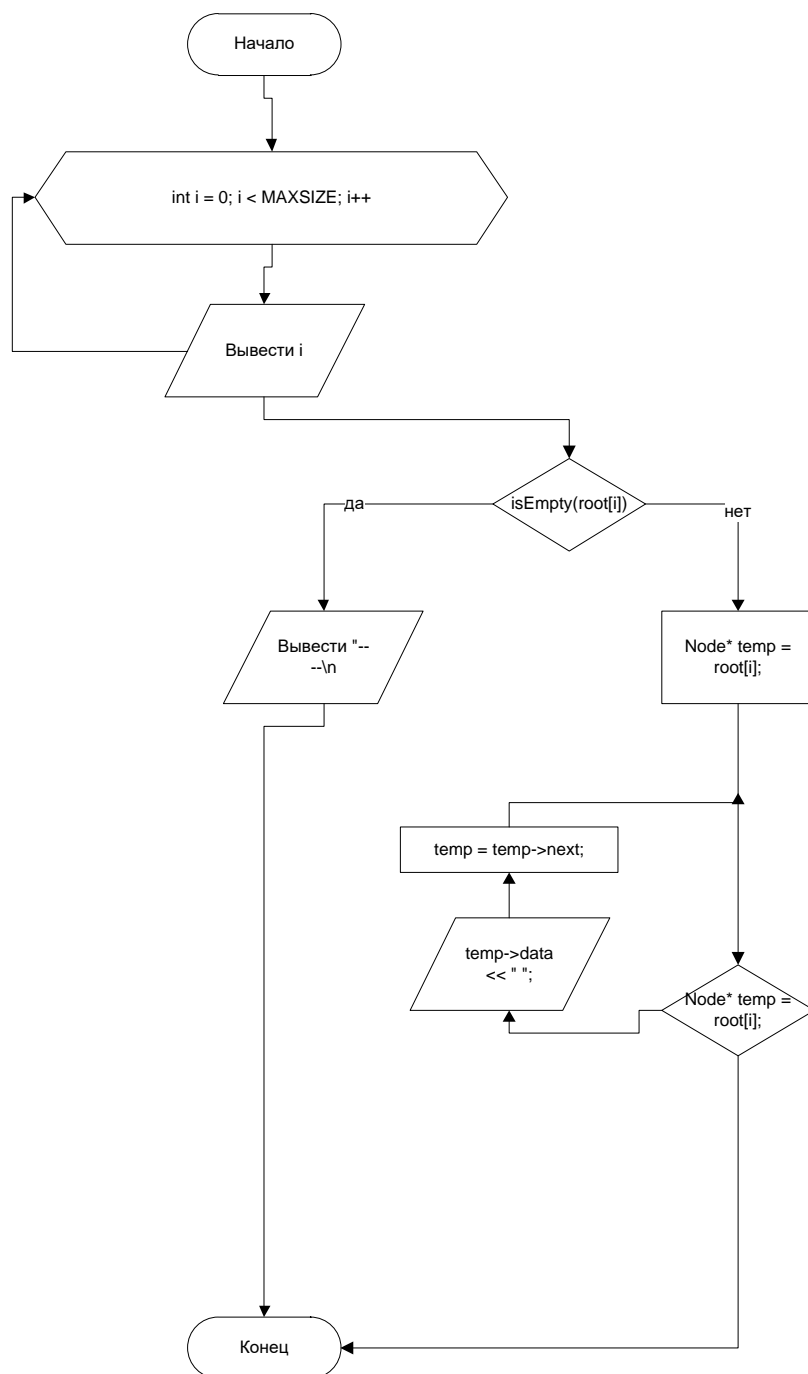


Рисунок 1 - Диаграмма деятельности для функции хеширования  
Добавление элемента в хеш-таблицу

При добавлении элементов в хеш-таблицу выделяются куски динамической памяти, которые организуются в виде связанных списков, каждый из которых соответствует входу хеш-таблицы.

//Добавление элемента в хеш - таблицу

```

void hashtable_add(struct listnode * *hashtab, char* key,
int value) {
    struct listnode* node;
    int index = hashtable_hash(key);
    //Вставка в начало списка
    node = malloc(sizeof(*node));
    if (node != NULL) {
        node->key = key;
        node->value = value;
        node->next = hashtab[index];
        hashtab[index] = node;
    }
}

```

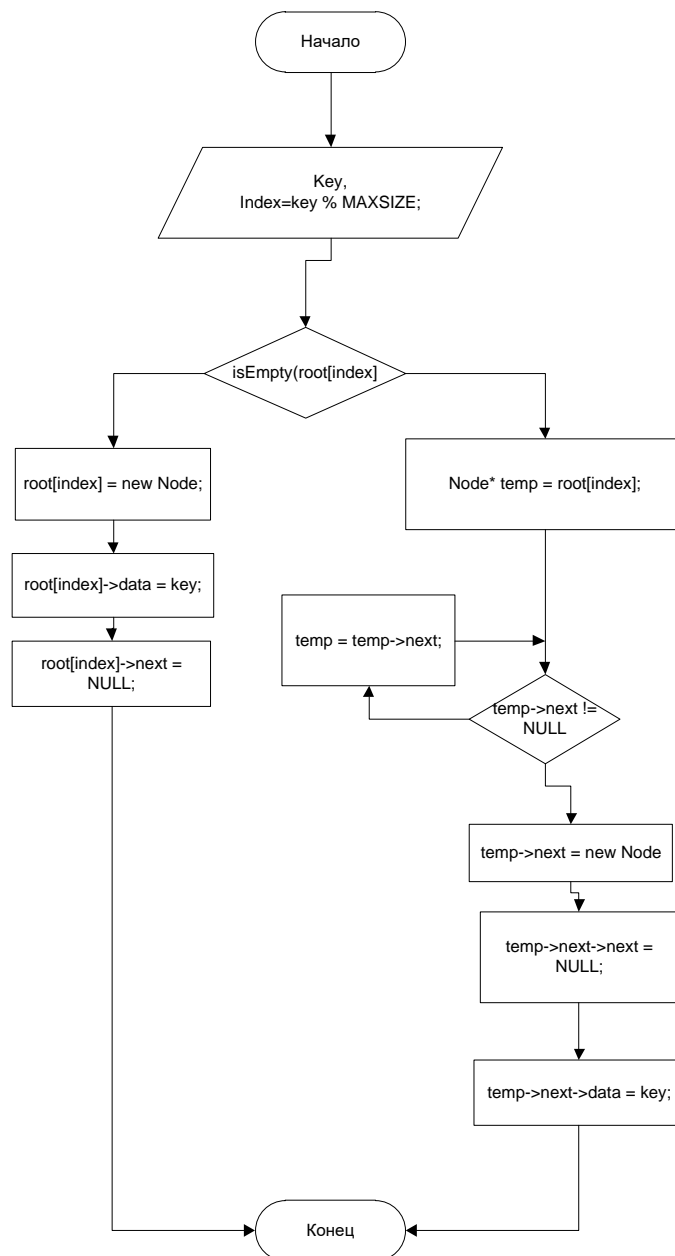


Рисунок 2 - Диаграмма деятельности для добавления элемента в таблицу

**Результат**

```
C:\Users\Asus\source\repos\Console/
-----
JUST A HASH TABLE
1-Insert
2-Delete
3-Print
4-Search
5-ANY KEY TO EXIT
Command:1

Enter Number to insert: 1

1-Insert
2-Delete
3-Print
4-Search
5-ANY KEY TO EXIT
Command:1

Enter Number to insert: 10
```

## Поиск элемента

```
//Поиск элемента
struct listnode* hashtable_lookup(
    struct listnode** hashtable, char* key);
{
    int index;
    struct listnode* node;
    index = hashtable_hash(key);
    for (node = hashtable[index];
        node != NULL; node = node->next)
    {
        if (strcmp(node->key, key) == 0)
            return node;
    }
    return NULL;
}
```

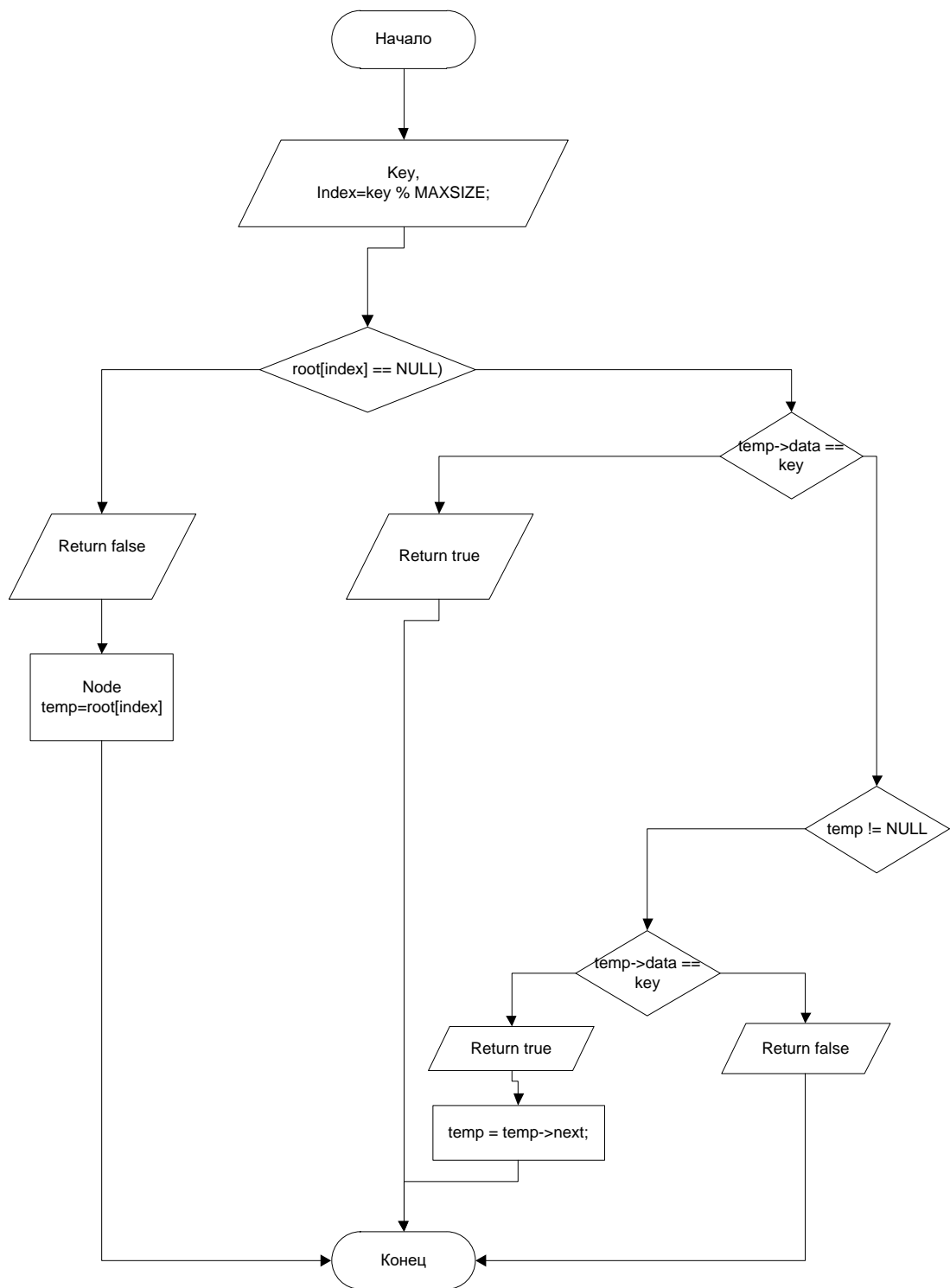


Рисунок 3 - Диаграмма деятельности поиска элемента

## Результат

```

Enter Number to Search: 1
1
1-Insert
2-Delete
3-Print
4-Search
5-ANY KEY TO EXIT
Command:4

Enter Number to Search: 2
0
1-Insert
2-Delete
3-Print
4-Search
5-ANY KEY TO EXIT
  
```

Изм.	Лист	№ докум.	Подпись	Дата

АуСД.09.03.02.050000 ПР

Лист

6

## Удаление элемента

//Удаление элемента

```
void hashtable_delete(struct listnode** hashtable, char* key) {
    int index;
    struct listnode* p, * prev = NULL;
    index = hashtable_hash(key);
    for (p = hashtable[index]; p != NULL; p = p->next) {
        if (strcmp(p->key, key) == 0) {
            {
                if (prev == NULLhashtable[index] = p->next;
                else
                prev->next = p->next;
                free(p);
                return; }
            prev = p; } }
}
```

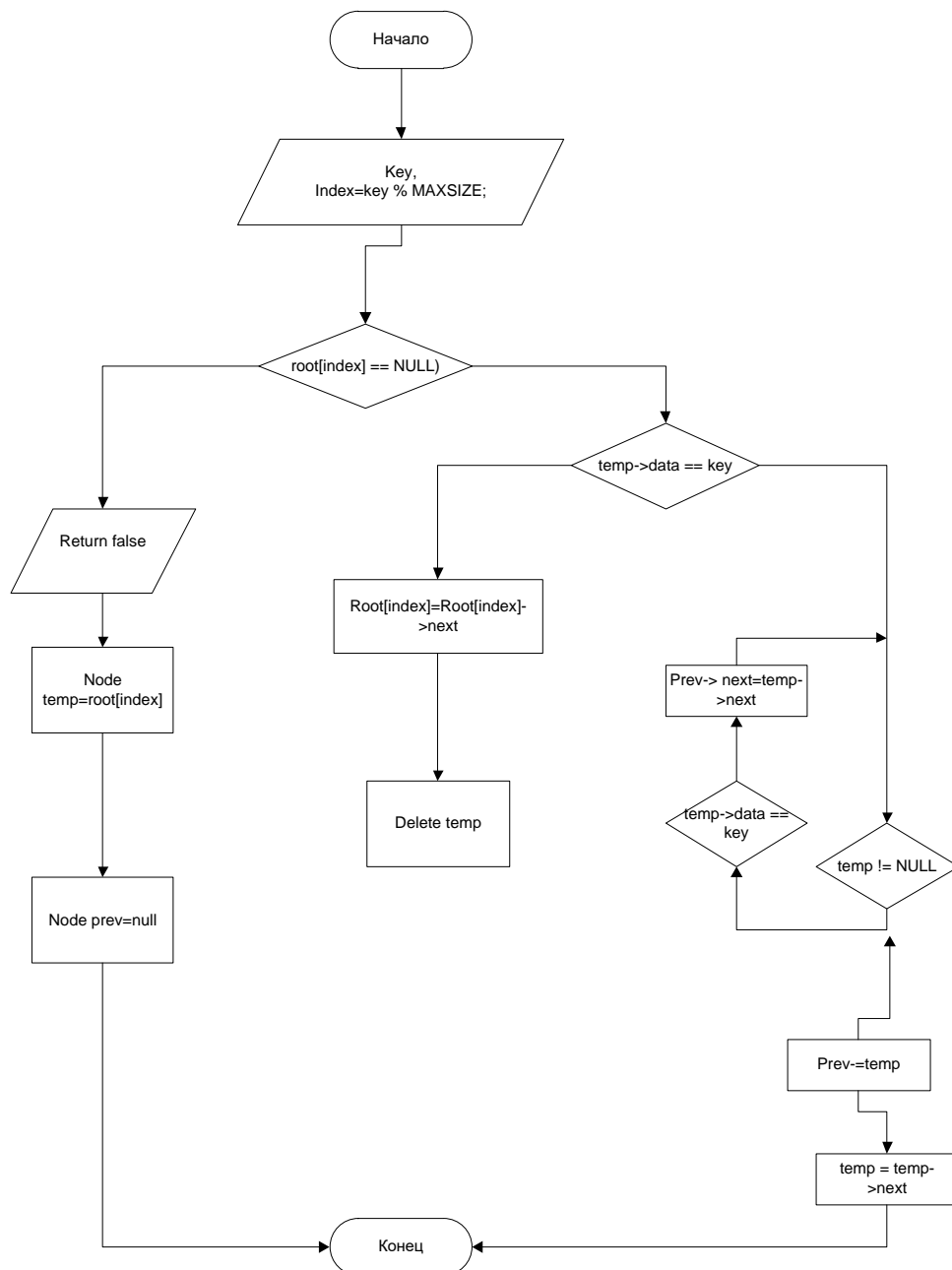


Рисунок 4 - Диаграмма деятельности удаления элемента

Изм.	Лист	№ докум.	Подпись	Дата

АИСД.09.03.02.050000 ПР

Лист

7

## Результат

```
Enter Number to delete: 1
1-Insert
2-Delete
3-Print
4-Search
5-ANY KEY TO EXIT
Command:3
0:      10
1:      ----
2:      ----
3:      ----
4:      ----
```

Вывод: в ходе выполнения лабораторной работы изучила алгоритмы хеширования таблиц.

					<i>АиСД.09.03.02.050000 ПР</i>	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		