ГУО «БГУИР»

Факультет информационной безопасности

Кафедра вычислительных методов и программирования

Отчет по

Лабораторной работе №7

Подготовил:

Студент гр.368402

Струнец А. П.

Проверила:

Семижон Е.А.

Минск 2024

Реализация хэш-таблицы на основе двухсвязных списков.

Код программы:

#include<iostream>

using namespace std;

struct list{

int data;

list\* next;

};

int getvalue(int left = 26000, int right = 77000);

list\*\* create(int M);

void add(list\*\* hash, int data, int M);

list\* search(list\*\* hash, int data, int hash\_size);

void clear(list\*\* hash, int hash\_size);

int main(){

setlocale(LC\_ALL, "rus");

int arr\_size;

cout << "Введите количество элементов массива: ";

cin >> arr\_size;

int\* arr = new int[arr\_size];

for (int i = 0; i < arr\_size; i++){

cout << "arr[" << i << "] = ";

arr[i] = getvalue(0);

}

int hash\_size;

cout << "Введите количество элементов в хэш-таблице: " << endl;

cin >> hash\_size;

list\*\* hash;

hash = create(hash\_size);

for (int i = 0; i < arr\_size; i++)

if (search(hash, arr[i], hash\_size) == NULL)

add(hash, arr[i], hash\_size);

delete[]arr;

list\* temp\_ptr;

int temp\_data = -1;

while (temp\_data != 0){

cout << "Введите элемент, который требуется найти (0 - стоп): ";

cin >> temp\_data;

if (temp\_data == 0) break;

temp\_ptr = search(hash, temp\_data, hash\_size);

if (temp\_ptr == NULL)

cout << "\nЭлемента в хэш-таблице нет" << endl;

else

cout << "\nЭлемент " << temp\_ptr->data << " в хэш-таблице присутствует" << endl;

}

clear(hash, hash\_size);

return 0;

}

int getvalue(int left, int right){

int value;

while (true)

{

cin >> value;

if (value >= left && value <= right) break;

cout << "Значение введено неверно. Повторите ввод" << endl;

}

return value;

}

list\*\* create(int hash\_size){

list\*\* hash = new list \* [hash\_size];

for (int i = 0; i < hash\_size; i++) hash[i] = NULL;

return hash;

}

void add(list\*\* hash, int data, int hash\_size){

list\* temp = new list;

temp->data = data;

int i = data % hash\_size;

if (hash[i] == NULL){

hash[i] = temp;

temp->next = NULL;

}

else{

temp->next = hash[i];

hash[i] = temp;

}

}

list\* search(list\*\* hash, int data, int hash\_size){

int i = abs(data % hash\_size);

list\* temp = hash[i];

while (temp != NULL){

if (temp->data == data) return temp;

temp = temp->next;

}

return NULL;

}

void clear(list\*\* hash, int hash\_size){

list\* del, \* temp;

for (int i = 0; i < hash\_size; i++){

cout << "H[" << i << "] = ";

temp = hash[i];

while (temp != NULL){

cout << temp->data << "->";

del = temp;

temp = temp->next;

delete del;

}

cout << "NULL" << endl;

}

delete[] hash;

}

Результат выполнения программы:

