Лабораторная работа 6

По дисциплине «Основы алгоритмизации и программирования»

На тему « Динамические структуры данных. Односвязные списки»

***Задание#1***



*Macros.h*

#include <iostream>

#include <vector>

#include <ctime>

#include <cmath>

#include <iomanip>

#include <stack>

#include <map>

#include <cstring>

#include <string>

#include <chrono>

#include <fstream>

#include <Windows.h>

#define vi cout <<

#define vv cin >>

#define nl cout << "\n";

#define nw cout << "\t";

#define SCOCP1251 SetConsoleOutputCP(1251);

#define SCCP1251 SetConsoleCP(1251);

#define ret return

#define pause system("pause");

using namespace :: std;

*.cpp*

#include "Macros.h"

// Структура для представления узла связанного списка

struct LinkedList {

char value;

LinkedList\* next;

};

// Прототипы функций

void insert(LinkedList\*& ptrNext);

void display(LinkedList\* ptr);

void search(LinkedList\* ptr);

int main() {

SCOCP1251

SCCP1251

LinkedList\* ptr = nullptr; // Указатель на текущий узел

int choice;

do {

nl vi "----- Меню -----"; nl

vi "1. Добавить символ"; nl

vi "2. Отобразить список"; nl

vi "3. Поиск символа"; nl

vi "0. Выйти"; nl

vi "Выберите действие: ";

vv choice;

switch (choice) {

case 1:

insert(ptr); // Добавление символа

break;

case 2:

display(ptr); // Отображение списка

break;

case 3:

search(ptr); // Поиск символа

break;

case 0:

vi "Выход...";

break;

default:

vi "Неправильный выбор. Пожалуйста, попробуйте снова.\n";

}

} while (choice != 0);

// Освобождение памяти, выделенной для узлов списка

while (ptr) {

LinkedList\* temp = ptr;

ptr = ptr->next;

delete temp;

}

pause

ret 0;

}

// Функция для добавления нового символа в начало связанного списка

void insert(LinkedList\*& ptrNext) {

LinkedList\* ptr;

vi "Введите символ: ";

char ch;

vv ch;

// Создаем новый узел и заполняем его данными

ptr = new LinkedList;

ptr->value = ch;

ptr->next = ptrNext; // Устанавливаем указатель на следующий элемент (переданный в функцию узел списка)

ptrNext = ptr; // Устанавливаем указатель на новый элемент в начало списка

}

// Функция для отображения символов связанного списка

void display(LinkedList\* ptr) {

if (ptr == nullptr) {

vi "Список пуст"; nl

}

while (ptr) {

vi ptr->value; nw // Выводим значение текущего элемента

ptr = ptr->next; // Переходим к следующему элементу

}

nl

}

// Функция для поиска символа в связанном списке

void search(LinkedList\* ptr) {

vi "Введите символ, который нужно найти: ";

char target;

vv target;

while (ptr) {

if (ptr->value == target) {

vi "Символ " << target << " найден в списке. Его следующий символ: ";

if (ptr->next) {

vi ptr->next->value; nl

}

else {

vi "отсутствует (конец списка)"; nl

}

ret;

}

ptr = ptr->next; // Переходим к следующему узлу

}

vi "Символ " << target << " не найден в списке"; nl

}

 