|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | |  | | | | | |
|  | | |
|  | Институт информационных технологий (ИТ) |
|  | Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ) |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3** | |
| **по дисциплине** | |
| **«Структуры и алгоритмы обработки данных»**  **По теме: «Рекурсивные функции»**  **Вариант 4** | |
|  | |
| Выполнил студент группы ИВБО-06-17 | Галеев Д.Р. |
| Принял преподаватель | Скворцова Л.А. |

Москва 2019

**Оглавление**

[1. Задание 1. Разработать функцию на рекурсии. 3](#_Toc27519085)

[1.1. Вариант задания 3](#_Toc27519086)

[1.2. Рекурентное соотношение 3](#_Toc27519087)

[1.3. Реализация рекурсивной функции 3](#_Toc27519088)

[10.1. Таблица тестов 3](#_Toc27519089)

[11. Задание 2. Разработать функцию на рекурсии, используя структуру списка. 3](#_Toc27519090)

[11.1. Вариант задания 3](#_Toc27519091)

[Удалить из связанного однонаправленного списка все элементы, равные 3](#_Toc27519092)

[заданному. 3](#_Toc27519093)

[11.2. Реализация рекурсивной функции 4](#_Toc27519094)

[2.4 Таблица тестов 4](#_Toc27519095)

[**2.5 Текст исходного кода (листинг) программы** 5](#_Toc27519096)

[Файл node.h 5](#_Toc27519097)

[Файл lab3z1.cpp 7](#_Toc27519098)

[**3.Задание 3. Разработать программу тестирования рекурсивных функций.** 8](#_Toc27519099)

[**3.1Вариант задания** 8](#_Toc27519100)

[3.2 Контрольные прогоны программы 8](#_Toc27519101)

1. Задание 1. Разработать функцию на рекурсии.
   1. Вариант задания

Определить, является ли текст – палиндромом.

Требования:

* 1. Описать рекурсивную зависимость в виде табличной функции
  2. Определить глубину рекурсии.
  3. Рекурентное соотношение

, где ai – итый символ строки a, aj – житый символ строки, i – индекс левой части, j – индекс правой части. Глубина рекурсии – j / 2.

* 1. Реализация рекурсивной функции

1. int is\_pal(string s, size\_t start, size\_t end)
2. {
3. if (start >= end)
4. return (1);
5. if (s[start] == s[end])
6. return (is\_pal(s, ++start, --end));
7. else
8. return (0);
9. }
   1. Таблица тестов

|  |  |  |
| --- | --- | --- |
| **Номер теста** | **Исходные данные** | **Эталон результата, глубина рекурсии** |
| **1** | АБВБА | 1, 2 |
| **2** | **1233321** | 1, 3 |
| **3** | **123421** | 0, 3 |

1. Задание 2. Разработать функцию на рекурсии, используя структуру списка.
   1. Вариант задания

Удалить из связанного однонаправленного списка все элементы, равные

заданному.

* 1. Реализация рекурсивной функции

1. t\_node \*del\_nodes(t\_node\* node, t\_node \*prev, int info)
2. {
3. if (!node)
4. return (NULL);
5. if (node->info == info)
6. {
7. t\_node\* tmp = node->next;
8. delete node;
9. return (del\_nodes(tmp, prev, info));
10. }
11. else
12. node->next = del\_nodes(node->next, node, info);
13. return (node);
14. }

2.4 Таблица тестов

|  |  |  |
| --- | --- | --- |
| **Номер теста** | **Исходные данные** | **Эталон результата** |
| **1** | **Для 1 элемента:**  **Введите информацию: 1**  **Для 2 элемента:**  **Введите информацию: 2**  **Для 3 элемента:**  **Введите информацию: 3**  **Для 4 элемента:**  **Введите информацию: 4**  **Введите информацию, по которой нужно удалить элементы: 3** | 1 2 4 |
| **2** | **Сколько элементов нужно создать: 3**  **Для 1 элемента:**  **Введите информацию: 1**  **Для 2 элемента:**  **Введите информацию: 2**  **Для 3 элемента:**  **Введите информацию: 2**  **1 2 2**  **Введите информацию, по которой нужно удалить элементы: 2** | 1 |
| **3** | **Cколько элементов нужно создать: 5**  **Для 1 элемента:**  **Введите информацию: 1**  **Для 2 элемента:**  **Введите информацию: 2**  **Для 3 элемента:**  **Введите информацию: 3**  **Для 4 элемента:**  **Введите информацию: 2**  **Для 5 элемента:**  **Введите информацию: 1**  **Введите информацию, по которой нужно удалить элементы: 1** | 2 3 2 |

**2.5 Текст исходного кода (листинг) программы**

Файл node.h

#include <iostream>

using namespace std;

typedef struct s\_node

{

int info;

struct s\_node\* next;

} t\_node;

t\_node\* node\_init(void)

{

t\_node\* res;

if (!(res = new t\_node))

exit(-1);

res->info = 0;

res->next = NULL;

return (res);

}

t\_node\* add\_new(int info)

{

t\_node\* res;

if (!(res = new t\_node))

exit(-1);

res->info = info;

res->next = NULL;

return (res);

}

t\_node\* creatennode(void)

{

t\_node\* head;

t\_node\* tmp;

size\_t i;

size\_t n;

int info;

std::cout << "Сколько элементов нужно создать: ";

std::cin >> n;

i = 0;

head = NULL;

tmp = NULL;

while (++i <= n)

{

if (!head)

{

head = node\_init();

std::cout << "Для " << i << " элемента:\n";

std::cout << "\tВведите информацию: ";

std::cin >> info;

head->info = info;

tmp = head;

}

else

{

std::cout << "Для " << i << " элемента:\n";

std::cout << "\tВведите информацию: ";

std::cin >> info;

tmp->next = add\_new(info);

tmp = tmp->next;

}

}

return (head);

}

void print\_node(t\_node\* head)

{

t\_node\* tmp;

tmp = head;

if (!head)

cout << "Нет элементов" << endl;

while (tmp)

{

cout << tmp->info;

if (!tmp->next)

cout << "\n";

else

cout << " ";

tmp = tmp->next;

}

}

Файл lab3z1.cpp

#include <iostream>

#include <string>

#include <stdlib.h>

#include "node.h"

using namespace std;

//Пример вызова функции head = del\_nodes(head, NULL, info);

t\_node \*del\_nodes(t\_node\* node, t\_node \*prev, int info)

{

if (!node)

return (NULL);

if (node->info == info)

{

t\_node\* tmp = node->next;

delete node;

return (del\_nodes(tmp, prev, info));

}

else

node->next = del\_nodes(node->next, node, info);

return (node);

}

//Пример вызова функции is\_pal(s, 0, s.length() - 1);

int is\_pal(string s, size\_t start, size\_t end)

{

if (start >= end)

return (1);

if (s[start] == s[end])

return (is\_pal(s, ++start, --end));

else

return (0);

}

int main()

{

setlocale(LC\_ALL, "RUS");

cout << "=============Демонстрация первого задания============" << endl;

string s;

cout << "Введите какую-либо строку, я проверю, является ли она палиндромом: ";

cin >> s;

int res = is\_pal(s, 0, s.length() - 1);

if (res)

cout << "Данная строка - палиндром" << endl;

else

cout << "Данная строка не является палиндромом" << endl;

cout << "=======================Конец=======================\n" << endl;

cout << "=============Демонстрация второго задания============" << endl;

t\_node\* head;

int info;

head = creatennode();

print\_node(head);

cout << "Введите информацию, по которой нужно удалить элементы: "; cin >> info;

head = del\_nodes(head, NULL, info);

print\_node(head);

cout << "=======================Конец=======================\n" << endl;

return (0);

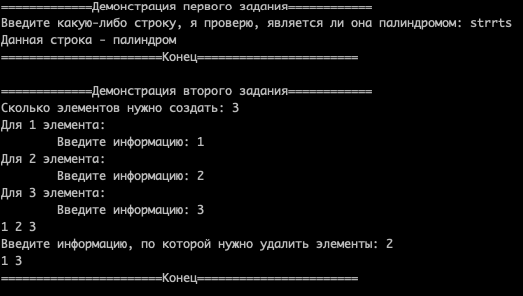
}

**3.Задание 3. Разработать программу тестирования рекурсивных функций.**

**3.1Вариант задания**

Разработать программу тестирования разработанных рекурсивных функций, которая управляется консольным меню.

* 1. Контрольные прогоны программы



Вывод

В результате выполнения лабораторной работы были получены навыки по созданию и работе с рекурсивными функциями.