

ГУАП

КАФЕДРА № 14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

П. В. Шпигун

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3

ЛИНЕЙНЫЕ СПИСКИ

по курсу: АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

1446

подпись

19.12.2025

подпись, дата

А. С. Пырву

инициалы, фамилия

Санкт-Петербург 2025

Цель работы: Изучить основные алгоритмы работы с линейными списками и освоить их на практике. Проверить работу алгоритмов на различных наборах данных.

Постановка задачи:

1. Разработать программу в соответствии с заданием к лабораторной работе.
2. Данные для построения списков вводить в режиме диалога.
3. Во всех работах разработать функцию удаления списков.

Вариант 21: Удалить из списка действительных чисел все максимальные элементы.

Формализация:

Константы:

EPS = 1e-6 - точность сравнения вещественных чисел

Размер списка для тестирования: 10 элементов

Реализованные функции:

list_create() - Создание нового узла списка

list_add() - Добавление элемента в конец списка

list_print() - Вывод списка на экран

list_free() - Освобождение памяти, занимаемой списком

double_equal() - Сравнение вещественных чисел с заданной точностью

list_delete() - Удаление всех элементов с заданным значением

list_max() - Поиск максимального элемента в списке

Использованные библиотеки:

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#include <time.h>

Листинг программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
#include <time.h>

#define EPS 1e-6

typedef struct _node node_t;

struct _node
{
    double value;
    node_t* next;
};

node_t* list_create()
{
    node_t* node = malloc(sizeof(node_t));
```

```

    node->next = NULL;

    return node;
}

void list_add(node_t** list, double value)
{
    node_t* node = list_create();
    node->value = value;

    if (*list == NULL)
    {
        *list = node;
        return;
    }

    node_t* tail = *list;

    while (tail->next != NULL)
    {
        tail = tail->next;
    }

    tail->next = node;
}

void list_print(node_t* list)
{
    while (list != NULL)
    {
        printf("(%p, %lf) -> %p\n", list, list->value, list->next);
        list = list->next;
    }
    printf("NULL\n");
}

void list_free(node_t** list)
{
    if (*list == NULL)
        return;

    node_t* next = (*list)->next;

    while (*list != NULL)
    {
        free(*list);

        *list = next;

        if (*list != NULL)
            next = (*list)->next;
    }

    *list = NULL;
}

bool double_equal(double a, double b)
{
    return fabs(a - b) < EPS;
}

void list_delete(node_t** list, double to_delete)
{
    while (*list != NULL && double_equal((*list)->value, to_delete))
    {

```

```

        node_t* tmp = (*list)->next;
        free(*list);
        *list = tmp;
    }

    if (*list == NULL)
        return;

    node_t* prev = *list;
    node_t* cur = prev->next;
    while (cur != NULL)
    {
        if (double_equal(cur->value, to_delete))
        {
            prev->next = cur->next;
            free(cur);
            cur = prev->next;
        }
        else
        {
            prev = cur;
            cur = cur->next;
        }
    }
}

double list_max(node_t* list)
{
    double m = list->value;
    while (list != NULL)
    {
        if (list->value > m)
        {
            m = list->value;
        }
        list = list->next;
    }

    return m;
}

int main()
{
    srand(time(NULL));

    node_t* list = NULL;

    for (int i = 0; i < 10; i++)
    {
        list_add(&list, rand() % 10);
    }

    list_print(list);

    double max_elem = list_max(list);

    list_delete(&list, max_elem);

    printf("max_elem: %lf\n", max_elem);

    list_print(list);
    list_free(&list);
}

```

Листинг 1. Код программы

Тестирование:

Консоль отладки Microsoft Visual Studio

```
(000001719C91E0A0, 8.000000) -> 000001719C91E5A0
(000001719C91E5A0, 2.000000) -> 000001719C91E550
(000001719C91E550, 3.000000) -> 000001719C91E500
(000001719C91E500, 5.000000) -> 000001719C91E0F0
(000001719C91E0F0, 1.000000) -> 000001719C91E5F0
(000001719C91E5F0, 1.000000) -> 000001719C91E190
(000001719C91E190, 9.000000) -> 000001719C91E280
(000001719C91E280, 9.000000) -> 000001719C91E640
(000001719C91E640, 3.000000) -> 000001719C91E1E0
(000001719C91E1E0, 6.000000) -> 0000000000000000
NULL
max_elem: 9.000000
(000001719C91E0A0, 8.000000) -> 000001719C91E5A0
(000001719C91E5A0, 2.000000) -> 000001719C91E550
(000001719C91E550, 3.000000) -> 000001719C91E500
(000001719C91E500, 5.000000) -> 000001719C91E0F0
(000001719C91E0F0, 1.000000) -> 000001719C91E5F0
(000001719C91E5F0, 1.000000) -> 000001719C91E640
(000001719C91E640, 3.000000) -> 000001719C91E1E0
(000001719C91E1E0, 6.000000) -> 0000000000000000
NULL
```

Рисунок 1. Результат удаления максимальных элементов из списка из 10-ти узлов.