

	<b>Отчёт по лабораторной работе №26</b> по курсу <b>1</b>
	студента группы <b>М8О-215Бв-24 Ярусов Ярослав</b> , № по списку <b>31</b>
	Адреса www, e-mail, jabber, skype <b>iaroslav</b>
	Работа выполнена: <b>“20” сентябрь 2025г.</b>
	Преподаватель: <b>Речинская Ангелина Юрьевна</b>
	Входной контроль знаний с оценкой _____
	Отчёт сдан <b>“29” сентября</b> _____ <b>2025</b> г., итоговая оценка _____
Подпись преподавателя _____	

☐ **Тема:** типы данных

☐ **Цель работы:** научиться реализовывать типы данных

☐ **Задание (вариант 1):** Линейный список, сортировка линейным выбором, удаление максимального элемента в списке

☐ **Оборудование (лабораторное):**

ЭВМ \_\_\_\_\_, процессор \_\_\_\_\_, имя узла сети \_\_\_\_\_ с ОП \_\_\_\_\_ МБ  
 НМД \_\_\_\_\_ ГБ. Терминал \_\_\_\_\_ адрес \_\_\_\_\_. Принтер \_\_\_\_\_  
 Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор **AMD Ryzen 7 4800U**, ОП **16** ГБ, НМД **1** ТБ. Монитор \_\_\_\_\_  
 Другие устройства \_\_\_\_\_

☐ **Программное обеспечение (лабораторное):**

Операционная система семейства \_\_\_\_\_, наименование \_\_\_\_\_ версия \_\_\_\_\_  
 Интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_  
 Система программирования \_\_\_\_\_ версия \_\_\_\_\_  
 Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_  
 Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_  
 Местонахождения и имена файлов программ и данных \_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства \_\_\_\_\_, наименование \_\_\_\_\_ версия \_\_\_\_\_  
 Интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_  
 Система программирования \_\_\_\_\_ версия \_\_\_\_\_  
 Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_  
 Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_  
 Местонахождения и имена файлов программ и данных \_\_\_\_\_

- **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

26main.c

```
#include "spisok.h"
#include "sort.h"
#include <stdio.h>

int main() {
    List lst, lstCopy, sorted, sorted2;
    InitList(&lst);
    InitList(&lstCopy);
    InitList(&sorted);
    InitList(&sorted2);

    int x;
    while (scanf("%d", &x) == 1) {
        Add(&lst, x);
        Add(&lstCopy, x);
        if (getchar() == '\n') break;
    }

    SelectionSort(&lst, &sorted);
    for (int i = 0; i < GetSize(&sorted); i++) printf("%d ", Get(&sorted, i));
    printf("\n");

    ExtractMax(&lstCopy);

    SelectionSort(&lstCopy, &sorted2);
    for (int i = 0; i < GetSize(&sorted2); i++) printf("%d ", Get(&sorted2, i));
    printf("\n");

    return 0;
}
```

spisok.c

```
#include "spisok.h"
#include <stdlib.h>

void InitList(List* lst) { lst->size = 0; }

bool IsEmpty(List* lst) { return lst->size == 0; }

bool IsFull(List* lst) { return lst->size >= MAX_SIZE; }

bool Add(List* lst, Element x) {
    if (IsFull(lst)) return false;
    lst->data[lst->size++] = x;
    return true;
}

bool RemoveAt(List* lst, int i, Element* x) {
    if (i < 0 || i >= lst->size) return false;
    *x = lst->data[i];
    for (int j = i; j < lst->size - 1; j++)
        lst->data[j] = lst->data[j + 1];
    lst->size--;
}
```

```

    return true;
}

int GetSize(List* lst) { return lst->size; }

Element Get(List* lst, int i) { return (i < 0 || i >= lst->size) ? -1 : lst->data[i]; }

```

spisok.h

```

#ifndef SPISOK_H
#define SPISOK_H

#include <stdbool.h>

#define MAX_SIZE 100

typedef int Element;

typedef struct {
    Element data[MAX_SIZE];
    int size;
} List;

void InitList(List* lst);
bool IsEmpty(List* lst);
bool IsFull(List* lst);
bool Add(List* lst, Element value);
bool RemoveAt(List* lst, int index, Element* value);
int GetSize(List* lst);
Element Get(List* lst, int index);

#endif

```

sort.c

```

#include "sort.h"

Element ExtractMax(List* lst) {
    int maxIndex = 0;
    Element maxValue = Get(lst, 0);
    for (int i = 1; i < GetSize(lst); i++) {
        Element v = Get(lst, i);
        if (v > maxValue) {
            maxValue = v;
            maxIndex = i;
        }
    }
    Element removed;
    RemoveAt(lst, maxIndex, &removed);
    return removed;
}

void SelectionSort(List* lst, List* sorted) {
    if (IsEmpty(lst)) return;
    Element x = ExtractMax(lst);
    Add(sorted, x);
    SelectionSort(lst, sorted);
}

```

sort.h

```
#ifndef SORT_H
#define SORT_H
```

```
#include "spisok.h"
```

```
Element ExtractMax(List* lst);
```

```
void SelectionSort(List* lst, List* sorted);
```

```
#endif
```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя \_\_\_\_\_

- ☐ **Распечатка протокола** (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)
- ☐ **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

--	--	--	--	--	--	--

☐ Замечание автора по существу работы \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

☐ Выводы : реализовал тип данных и сортировку

☐ Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Подпись студента \_\_\_\_\_