**Отчет по лабораторной работе № 26** по курсу алгоритмы и структуры данных

Студент группы М8О-101БВ-24 Волков Алексей Александрович, № по списку 3

Контакты e-mail

Работа выполнена: «22.04» 2025 г.

Преподаватель: \_\_\_\_\_\_\_ каф. 806 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_\_\_\_\_\_\_202 \_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Тема:** Абстрактные типы данных. Рекурсия. Модульное программирование на языке Си.

1. **Цель работы:**

Ознакомиться с типами данных: стек, дек, очередь, линейный список. Научиться составлять модули определений и реализации для этих типов данных. Ознакомиться со стандартными видами сортировок для предложенных типов данных, научиться реализовывать соответствующие модули.

1. **Задание** (*вариант №*)**:**

Составить и отладить модуль определений и модуль реализации для дека. Составить модули, сортирующий дек методом линейной выборки, используя только операции, импортированные из модуля дека.

1. **Оборудование** (лабораторное):

ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , процессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, имя узла сети \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с ОП \_\_\_\_\_\_\_\_\_\_ Мб, НМД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мб. Терминал \_\_\_\_\_ адрес \_\_\_\_\_\_\_\_\_\_. Принтер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с ОП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ГБ, НМД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ГБ. Монитор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Программное обеспечение (лабораторное):**

Операционная система семейства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, наименование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

интерпретатор команд \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение 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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**Дек** (двусторонняя очередь) на основе двусвязного списка. Каждый элемент дека представлен узлом, содержащим значение и два указателя — на следующий и предыдущий элементы. Дек хранит указатели на первый и последний элементы, что позволяет эффективно выполнять все основные операции.

Операции на деком:

1. Создание дека (*create\_dyn\_deque*).
2. Удаление дека с освобождением памяти (*destroy\_dyn\_deque*).
3. Проверка дека на пустоту (*is\_empty\_dyn\_deque*).
4. Добавление элемента в начало (*push\_front\_dyn\_deque*) и в конец (*push\_back\_dyn\_deque*).
5. Удаление элемента из начала (*pop\_front\_dyn\_deque*) и из конца (*pop\_back\_dyn\_deque*).

Для сортировки дека реализован **метод линейной выборки** (*selection* *sort*), использующий только основные операции дека. Сортировка выполняется следующим образом: из дека последовательно выбирается максимальный элемент (с сохранением остальных во вспомогательном деке), найденный максимум переносится в результирующую структуру. Затем элементы возвращаются обратно — в итоге дек оказывается отсортирован по неубыванию.

Алгоритм:

1. Ввод исходных данных в дек
2. Пока дек не пуст:
   1. Найти максимальный элемент с помощью *findMax*
   2. Переместить максимум во вспомогательный дек
3. Перенести элементы из вспомогательного дека обратно в исходный (сортировка по возрастанию)
4. Вывести результат
5. Освободить память, завершить программу

**7. Сценарий выполнения работы** [план работы,первоначальный текст программы в черновике(можно на отдельном листе)итесты либо соображения по тестированию].

**1. Описана структура** *dyn\_deque***, включающая указатели на первый и последний элементы.**

1. Для представления элементов дека создана структура elem, содержащая значение (тип *double*) и указатели на следующий и предыдущий элементы.
2. Реализованы основные операции с деком:
   1. создание (*create\_dyn\_deque*);
   2. удаление с освобождением памяти (*destroy\_dyn\_deque*);
   3. проверка на пустоту (*is\_empty\_dyn\_deque*);
   4. добавление элементов в начало и конец (*push\_front\_dyn\_deque*, *push\_back\_dyn\_deque*);
   5. удаление элементов с обоих концов (*pop\_front\_dyn\_deque*, *pop\_back\_dyn\_deque*).

**2. Разработка модуля сортировки дека методом линейной выборки:**

1. Описана и реализована функция поиска максимального элемента (*findMax*), использующая только операции дека. При этом максимальный элемент удаляется из дека, остальные элементы сохраняются во вспомогательной структуре.
2. Создана функция сортировки (*choose\_sort*), которая последовательно извлекает максимумы из исходного дека и переносит их в новый, обеспечивая сортировку по не убыванию.

**3. Проведение тестирования реализованных функций:**

**Тест 1. Пустой дек**

Входные данные: Пусто (нет элементов на входе)

Ожидаемый результат: На выходе также пусто, программа завершается без ошибок.

Фактический результат: После сортировки дек остаётся пустым, ошибок не возникает.

**Тест 2. Один элемент**

Входные данные: 5

Ожидаемый результат: 5

Фактический результат: После сортировки дек содержит тот же элемент (5).

**Тест 3. Несколько случайных чисел**

Входные данные: 4 1 7 3

Ожидаемый результат: 1 3 4 7

Фактический результат: После сортировки элементы выведены в порядке возрастания: 1 3 4 7.

**Тест 4. Повторяющиеся значения**

Входные данные: 2 5 2 3 5

Ожидаемый результат: 2 2 3 5 5

Фактический результат: Сортировка корректно размещает одинаковые элементы рядом: 2 2 3 5 5.

**Тест 5. Уже отсортированный по возрастанию дек**

Входные данные: 1 2 3 4 5

Ожидаемый результат: 1 2 3 4 5

Фактический результат: После сортировки порядок не меняется: 1 2 3 4 5.

**Тест 6. Отсортированный по убыванию дек**

Входные данные: 9 7 5 3 1

Ожидаемый результат: 1 3 5 7 9

Фактический результат: Сортировка изменяет порядок на возрастающий: 1 3 5 7 9.

**Тест 7. Отрицательные и положительные значения**

Входные данные: -3 7 0 -1 5

Ожидаемый результат: -3 -1 0 5 7

Фактический результат: После сортировки числа упорядочены: -3 -1 0 5 7.

**Тест 8. Большое количество элементов**

Входные данные: Вводится последовательность из 150 случайных чисел.

Ожидаемый результат: Все числа отсортированы по возрастанию, программа завершается успешно, без утечек памяти.

Фактический результат: Сортировка выполняется корректно, ошибок и утечек памяти не обнаружено.

**4. Проверка устойчивости работы и корректности освобождения памяти:**

1. Контроль за отсутствием утечек памяти при создании и уничтожении деков, а также при выполнении операций удаления элементов.
2. При проведении тестов программа успешно проходила тестирование с различными сценариями, включая граничные случаи (пустой дек, один элемент, большие объёмы данных).

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

*Допущен к выполнению работы.*

**Подпись преподавателя**

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем)

**9. Дневник отладки** должен содержать дату и время сеансов отладки, и основные события(ошибки в сценарии и программе,нестандартные ситуации) 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. | Дата | Время | Событие | Действие по исправлению | Примечание |
| --- | --- | --- | --- | --- | --- | --- |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы

1. **Выводы**

* В ходе работы была успешно реализована структура дека с основными операциями и проведена сортировка методом выбора с помощью модульного подхода в языке C.
* Использование абстрактного типа данных позволило изолировать внутреннюю реализацию от алгоритмов сортировки, что улучшило читаемость и расширяемость кода.
* Проверка на разных наборах данных подтвердила корректность и устойчивость программы, а тщательное управление памятью обеспечило отсутствие утечек даже при интенсивной работе с динамическими структурами.
* Лабораторная работа показала, что подходы к абстракции и модульному программированию существенно упрощают разработку и сопровождение сложных алгоритмов в ЯП Cи.

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_