**Отчёт по лабораторной работе №25-26**

по курсу «Языки и методы программирования».

Выполнил студент группы **М8О-114БВ-24**: **Дылдин Сергей Владиславович** № по списку **10**.

Контакты: [**dsergey010206@gmail.com**](mailto:dsergey010206@gmail.com)

Работа выполнена: «28» мая 2025 г.

Преподаватель: **каф. 806 Никулин Сергей Петрович**

Входной контроль знаний с оценкой: **\_\_\_\_\_**

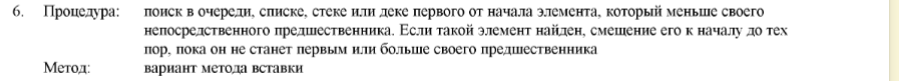
Отчет сдан «30» мая 2025 г.

Итоговая оценка: \_\_\_\_\_

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

1. **Тема:** динамические структуры данных, обработка деревьев.

2. **Цель работы:** составить программу на языке Си для построения и обработки дерева общего вида, содрежащего узлы типа int.

3. **Задание: 4.Линейный список** .

4. **Оборудование**: *Оборудование ПЭВМ студента, если использовалось:*

Процессор **Intel core i5 12400H**, ОП **16** ГБ, SSD **1**TB.

5. **Программное обеспечение:** *Программное обеспечение ПЭВМ студента, если использовалось****:***

Операционная система семейства **Ubuntu**, наименование версия **VirtualBox Ubuntu 20.04.3 LTS**,

интерпретатор команд **bash** версия **5.0.17**. Система программирования **C**.Редактор текстов **VI** версия **8.1**

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

*Составление программы на Си, выполняющей следующие задачи:*Инициализация списка:  
 При запуске программы вызывается функция list\_create, которая устанавливает указатель на начало списка (head) в NULL, тем самым создавая пустой список.

1. Вывод меню и ожидание ввода пользователя:  
    Функция print\_menu выводит набор доступных действий (например, добавить элемент в начало или конец, удалить первый или последний элемент, вставить или удалить по индексу, вывести список, вычислить его размер, отсортировать список или выйти из программы). Программа затем ожидает ввода пользователя.
2. Обработка выбора пользователя:  
    В бесконечном цикле в функции main ввод пользователя обрабатывается через оператор switch. В зависимости от выбранной опции программа:
   * Запрашивает дополнительные данные (значение или индекс);
   * Вызывает соответствующую функцию для выполнения операции над списком (например, list\_push\_front, list\_pop\_back, list\_insert и т.д.).
3. Выполнение операций над списком:  
    Каждая операция (добавление, удаление, вставка, сортировка) реализована в отдельной функции:
   * Добавление:  
      Функции list\_push\_front и list\_push\_back создают новый узел, выделяя память, и добавляют его в начало или конец списка.
   * Удаление:  
      Функции list\_pop\_front и list\_pop\_back удаляют соответствующие узлы, освобождая память, а list\_erase удаляет узел по указанному индексу.
   * Вставка:  
      Функция list\_insert вставляет новый узел в нужную позицию, корректируя связи между узлами.
   * Вывод и подсчет:  
      Функция list\_print выводит все элементы списка, а list\_size подсчитывает их количество.
   * Сортировка:  
      Функция list\_insertion\_sort сортирует список методом вставок, создавая новый отсортированный список и переназначая указатель head.
4. Завершение работы:  
    Если пользователь выбирает опцию выхода (например, вводит «0»), цикл завершается, и программа завершает работу.

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

linked\_list.h

#ifndef SINGLY\_LINKED\_LIST\_H

#define SINGLY\_LINKED\_LIST\_H

#include <stdbool.h> // Для типа bool

#include <stddef.h> // Для типа size\_t

// Определение структуры узла списка

typedef struct Node {

int data;

struct Node\* next;

} Node;

// Определение структуры списка

typedef struct {

Node\* head;

} SinglyLinkedList;

// Операции над списком

void list\_create(SinglyLinkedList \*);

bool list\_is\_empty(const SinglyLinkedList \*);

void list\_push\_front(SinglyLinkedList \*, int);

void list\_push\_back(SinglyLinkedList \*, int);

void list\_pop\_front(SinglyLinkedList \*);

void list\_pop\_back(SinglyLinkedList \*);

void list\_print(const SinglyLinkedList \*);

size\_t list\_size(const SinglyLinkedList \*);

void list\_insert(SinglyLinkedList \*, size\_t, int);

void list\_erase(SinglyLinkedList \*, size\_t);

void list\_insertion\_sort(SinglyLinkedList \*list);

#endif // SINGLY\_LINKED\_LIST\_H

**oper.c**

#include "linked\_list.h"

#include <stdio.h>

#include <stdlib.h>

// Инициализация списка

void list\_create(SinglyLinkedList \*list) {

list->head = NULL;

}

// Проверка, пуст ли список

bool list\_is\_empty(const SinglyLinkedList \*list) {

return list->head == NULL;

}

// Добавление элемента в начало

void list\_push\_front(SinglyLinkedList \*list, int value) {

Node \*new\_node = (Node \*)malloc(sizeof(Node));

if (!new\_node) return;

new\_node->data = value;

new\_node->next = list->head;

list->head = new\_node;

}

// Добавление элемента в конец

void list\_push\_back(SinglyLinkedList \*list, int value) {

Node \*new\_node = (Node \*)malloc(sizeof(Node));

if (!new\_node) return;

new\_node->data = value;

new\_node->next = NULL;

if (list\_is\_empty(list)) {

list->head = new\_node;

return;

}

Node \*temp = list->head;

while (temp->next) {

temp = temp->next;

}

temp->next = new\_node;

}

// Удаление первого элемента

void list\_pop\_front(SinglyLinkedList \*list) {

if (list\_is\_empty(list)) return;

Node \*temp = list->head;

list->head = temp->next;

free(temp);

}

// Удаление последнего элемента

void list\_pop\_back(SinglyLinkedList \*list) {

if (list\_is\_empty(list)) return;

if (!list->head->next) {

free(list->head);

list->head = NULL;

return;

}

Node \*temp = list->head;

while (temp->next->next) {

temp = temp->next;

}

free(temp->next);

temp->next = NULL;

}

// Вывод списка

void list\_print(const SinglyLinkedList \*list) {

Node \*temp = list->head;

while (temp) {

printf("%d -> ", temp->data);

temp = temp->next;

}

printf("NULL\n");

}

// Получение размера списка

size\_t list\_size(const SinglyLinkedList \*list) {

size\_t count = 0;

Node \*temp = list->head;

while (temp) {

count++;

temp = temp->next;

}

return count;

}

// Вставка элемента по индексу

void list\_insert(SinglyLinkedList \*list, size\_t index, int value) {

if (index == 0) {

list\_push\_front(list, value);

return;

}

Node \*temp = list->head;

for (size\_t i = 0; temp && i < index - 1; i++) {

temp = temp->next;

}

if (!temp) return;

Node \*new\_node = (Node \*)malloc(sizeof(Node));

if (!new\_node) return;

new\_node->data = value;

new\_node->next = temp->next;

temp->next = new\_node;

}

// Удаление элемента по индексу

void list\_erase(SinglyLinkedList \*list, size\_t index) {

if (list\_is\_empty(list)) return;

if (index == 0) {

list\_pop\_front(list);

return;

}

Node \*temp = list->head;

for (size\_t i = 0; temp->next && i < index - 1; i++) {

temp = temp->next;

}

if (!temp->next) return;

Node \*to\_delete = temp->next;

temp->next = to\_delete->next;

free(to\_delete);

}

// Сортировка списка вставками

void list\_insertion\_sort(SinglyLinkedList \*list) {

if (!list->head || !list->head->next) return;

Node \*sorted = NULL;

Node \*current = list->head;

while (current) {

Node \*next = current->next;

if (!sorted || sorted->data >= current->data) {

current->next = sorted;

sorted = current;

} else {

Node \*temp = sorted;

while (temp->next && temp->next->data < current->data) {

temp = temp->next;

}

current->next = temp->next;

temp->next = current;

}

current = next;

}

list->head = sorted;

}

**main.c**

#include "linked\_list.h"

#include <stdio.h>

#include <stdlib.h>

void print\_menu() {

printf("\nМеню:\n");

printf("1. Добавить элемент в начало\n");

printf("2. Добавить элемент в конец\n");

printf("3. Удалить первый элемент\n");

printf("4. Удалить последний элемент\n");

printf("5. Вставить элемент по индексу\n");

printf("6. Удалить элемент по индексу\n");

printf("7. Вывести список\n");

printf("8. Вывести размер списка\n");

printf("9. Отсортировать список\n");

printf("0. Выход\n");

printf("Выберите действие: ");

}

int main() {

SinglyLinkedList list;

list\_create(&list);

int choice, value, index;

while (1) {

print\_menu();

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Введите значение: ");

scanf("%d", &value);

list\_push\_front(&list, value);

break;

case 2:

printf("Введите значение: ");

scanf("%d", &value);

list\_push\_back(&list, value);

break;

case 3:

list\_pop\_front(&list);

break;

case 4:

list\_pop\_back(&list);

break;

case 5:

printf("Введите индекс и значение: ");

scanf("%d %d", &index, &value);

list\_insert(&list, index, value);

break;

case 6:

printf("Введите индекс: ");

scanf("%d", &index);

list\_erase(&list, index);

break;

case 7:

list\_print(&list);

break;

case 8:

printf("Размер списка: %zu\n", list\_size(&list));

break;

case 9:

printf("9. Отсортировать список по возрастанию:\n");

list\_insertion\_sort(&list);

break;

case 0:

return 0;

default:

printf("Некорректный ввод, попробуйте снова.\n");

}

}

}

**Makefile**

CC = gcc

CFLAGS = -Wall -Wextra -std=c11

SRC = main.c oper.c

OBJ = main.o oper.o

EXEC = program.exe

all: $(EXEC)

$(EXEC): $(OBJ)

$(CC) $(CFLAGS) -o $(EXEC) $(OBJ)

main.o: main.c linked\_list.h

$(CC) $(CFLAGS) -c main.c -o main.o

oper.o: oper.c linked\_list.h

$(CC) $(CFLAGS) -c oper.c -o oper.o

clean:

-del /F /Q $(OBJ) $(EXEC) 2> NUL

*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, подписанный преподавателем).*

PS D:\apps\labi\2sem\laba4> make

gcc -Wall -Wextra -std=c11 -c main.c -o main.o

gcc -Wall -Wextra -std=c11 -c oper.c -o oper.o

gcc -Wall -Wextra -std=c11 -o program.exe main.o oper.o

PS D:\apps\labi\2sem\laba4> ls

Directory: D:\apps\labi\2sem\laba4

Mode LastWriteTime Length Name

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

-a---- 16.03.2025 22:27 997 linked\_list.h

-a---- 30.03.2025 21:25 2537 main.c

-a---- 30.03.2025 22:20 3501 main.o

-a---- 30.03.2025 21:21 363 Makefile

-a---- 16.03.2025 22:25 4021 oper.c

-a---- 30.03.2025 22:20 3092 oper.o

-a---- 30.03.2025 22:20 143571 program.exe

PS D:\apps\labi\2sem\laba4> ./program.exe

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 8

Размер списка: 0

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 1

Введите значение: 10

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 1

Введите значение: 20

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 2

Введите значение: 30

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 1

Введите значение: 100

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 2

Введите значение: 10

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

100 -> 20 -> 10 -> 30 -> 10 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 9

9. Отсортировать список по возрастанию:

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

10 -> 10 -> 20 -> 30 -> 100 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 3

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 4

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

10 -> 20 -> 30 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 6

Введите индекс: 1

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

10 -> 30 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 5

Введите индекс и значение: 1 150

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

10 -> 150 -> 30 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 1

Введите значение: 23

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 2

Введите значение: 45

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 2

Введите значение: 1

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 8

Размер списка: 6

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

23 -> 10 -> 150 -> 30 -> 45 -> 1 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 9

9. Отсортировать список по возрастанию:

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

1 -> 10 -> 23 -> 30 -> 45 -> 150 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 8

Размер списка: 6

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 3

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 4

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 6

Введите индекс: 3

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

10 -> 23 -> 30 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 6

Введите индекс: 2

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

10 -> 23 -> NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 3

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 4

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 3

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 9

9. Отсортировать список по возрастанию:

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 8

Размер списка: 0

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 4

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 3

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 7

NULL

Меню:

1. Добавить элемент в начало

2. Добавить элемент в конец

3. Удалить первый элемент

4. Удалить последний элемент

5. Вставить элемент по индексу

6. Удалить элемент по индексу

7. Вывести список

8. Вывести размер списка

9. Отсортировать список

0. Выход

Выберите действие: 0

PS D:\apps\labi\2sem\laba4>

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. или дом.** | **Дата** | **Время** | **Событие** | **Действие по исправлению** | **Примечание** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

10. **Замечания автора** по существу работы: замечания отсутствуют.

11. **Выводы:** научился работать с деревьями общего вида в СП Си.

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