МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчёт по лабораторной работе №4
по дисциплине
«Программирование»
по теме: "Динамические структуры данных"

Выполнил студент гр. ИВТб-1301-06-00 _____ /Габдулбариев Т.Р./
Проверил доцент кафедры ЭВМ _____ /Долженкова М.Л./

Киров
2025

Цель

Цель лабораторной работы: Изучение механизмов эффективного управления динамической памятью, создание элементарных структур данных. Реализация дека.

Задание

Разработать консольное приложение для демонстрации особенностей организации элементарных структур данных. Для организации взаимодействия с пользователем использовать case-меню.

Исходный код программы для решения задачи на языке C:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>

#define UP 72
#define DOWN 80
#define ENTER 13

typedef struct Node
{
    int data;
    char datatwo[10];
    struct Node* next;
}
Node;
typedef struct Deque {
```

```
    Node* head = NULL;
    Node* tail = NULL;
}
Deque;

int global_length = 0;

void push_front(int value, char name[10], Deque* a)
{

    Node* n = (Node*)malloc(sizeof(Node));
    n->data = value;
    strncpy_s(n->datatwo, sizeof(n->datatwo), name, _TRUNCATE);

    n->next = a->head;
    a->head = n;

    if (a->tail == NULL) a->tail = n;
    global_length = global_length + 1;
}

void push_back(int value, char name[10], Deque* a)
{

    Node* n = (Node*)malloc(sizeof(Node));
    n->data = value;
    strncpy_s(n->datatwo, sizeof(n->datatwo), name, _TRUNCATE);
    n->next = NULL;

    if (a->tail)
```

```
    {
        a->tail->next = n;
        a->tail = n;
    }
    else
    {
        a->head = a->tail = n;
    }
    global_length = global_length + 1;
}

void pop_front(Deque* a)
{

    int key;
    if (!(a->head)) { printf("Deque is empty\n"); key = _getch(); if (key

    Node* n = a->head;
    a->head = a->head->next;
    free(n);
    if (!(a->head))
        a->tail = NULL;
    if (global_length != 0) global_length = global_length - 1;
}

void pop_back(Deque* a)
{

    int key;
    if (!(a->head)) { printf("Deque is empty\n"); key = _getch(); if (key
```

```
    if (a->head == a->tail)
    {
        free(a->head);
        a->head = a->tail = NULL;
        return;
    }

    Node* n = a->head;
    while (n->next != a->tail) n = n->next;

    free(a->tail);
    a->tail = n;
    a->tail->next = NULL;
    if (global_length != 0) global_length = global_length - 1;
}

void printDeque(Deque* a)
{

    if (!(a->head)) { printf("Deque is empty\n"); return; }
    printf("length: %d \n", global_length);
    printf("Deque: ");
    Node* p = a->head;
    while (p)
    {
        printf("(%d, %s) ", p->data, p->datatwo);
        p = p->next;
    }
    printf("\n");
}
```

```
void menu(Deque* a)
{
    const char* items[] = {
        "Add to front",
        "Add to back",
        "Remove from front",
        "Remove from back",
        "Exit"
    };

    int choice = 0;
    int key;
    char namecur[10];

    while (1)
    {
        system("cls");
        printf("=== DEQUE ===\n\n");
        printf("Two information fields: name and integer\n\n");
        printDeque(a);
        for (int i = 0; i < 5; i++)
        {
            if (i == choice)
                printf("-> %s\n", items[i]);
            else
                printf("   %s\n", items[i]);
        }
        key = _getch();
        if (key == 224)
```

```
{
    key = _getch();
    if (key == UP && choice > 0) choice--;
    else if (key == UP && choice == 0) choice = 4;
    if (key == DOWN && choice < 4) choice++;
    else if (key == DOWN && choice == 4) choice = 0;
}
else if (key == ENTER)
{
    system("cls");

    if (choice == 0)
    {
        int x;

        printf("Enter value: ");
        while (scanf_s("%d", &x) != 1)
        {
            printf("Error! Enter a number: ");
            int c; while ((c = getchar()) != '\n');
        }
        printf("Enter name: ");

        scanf_s("%9s", namecur, (unsigned)_countof(namecur));
        push_front(x, namecur, a);
    }
    else if (choice == 1)
    {
        int x;
```

```
                printf("Enter value: ");
                while (scanf_s("%d", &x) != 1)
                {
                    printf("Error! Enter a number: ");
                    int c; while ((c = getchar()) != '\n');
                }
                printf("Enter name: ");
                scanf_s("%9s", namecur, (unsigned)_countof(namecur));
                push_back(x, namecur, a);
            }
            else if (choice == 2) pop_front(a);
            else if (choice == 3) pop_back(a);
            else if (choice == 4) { return; }
        }
    }
}
int main()
{
    Deque* a = NULL;
    a = (Deque*)malloc(sizeof(Deque));
    if (a != NULL) {
        a->head = NULL;
        a->tail = NULL;
    }
    menu(a);
    return 0;
}
```

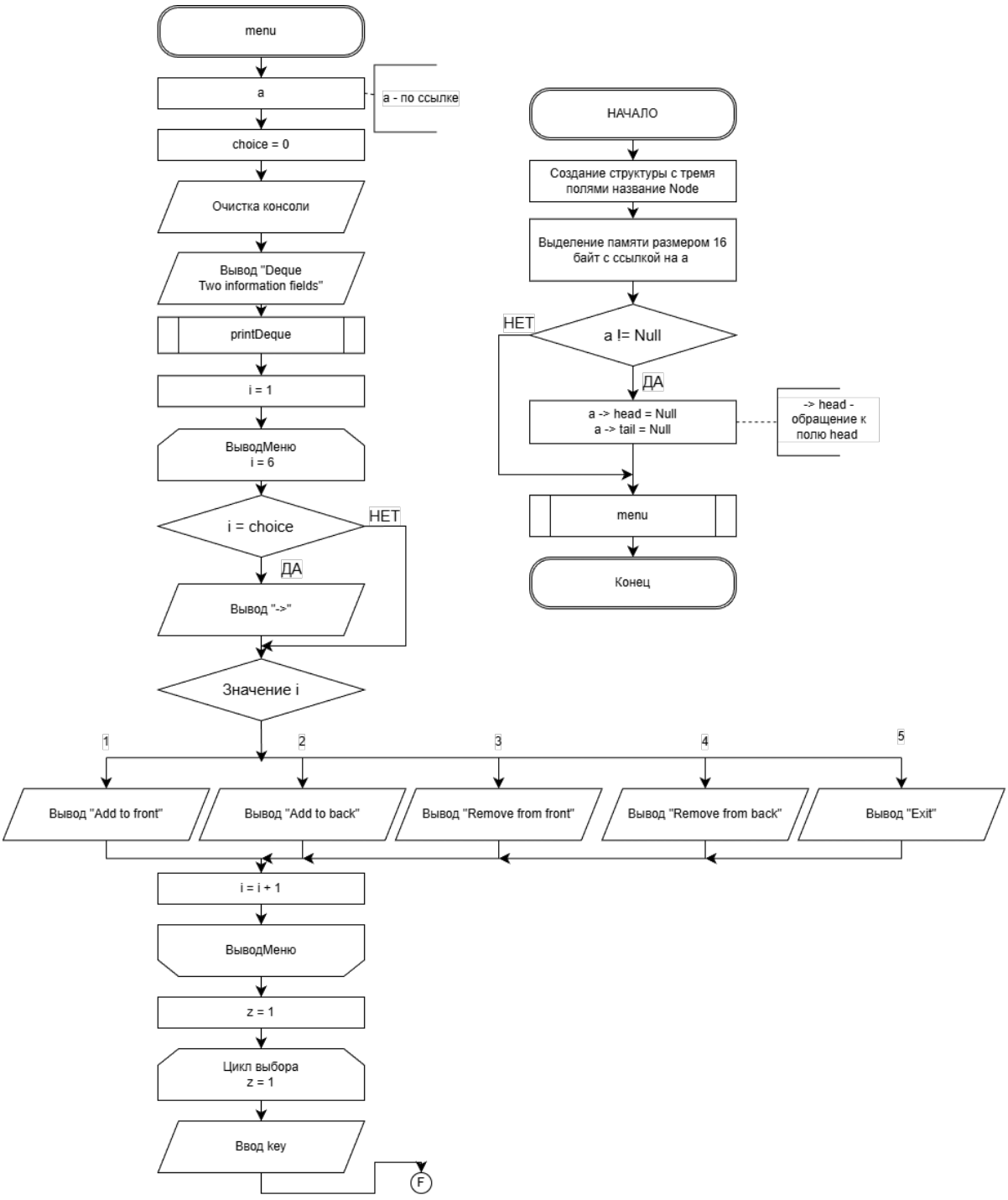

Рис. 1: Первая часть меню и основная программа

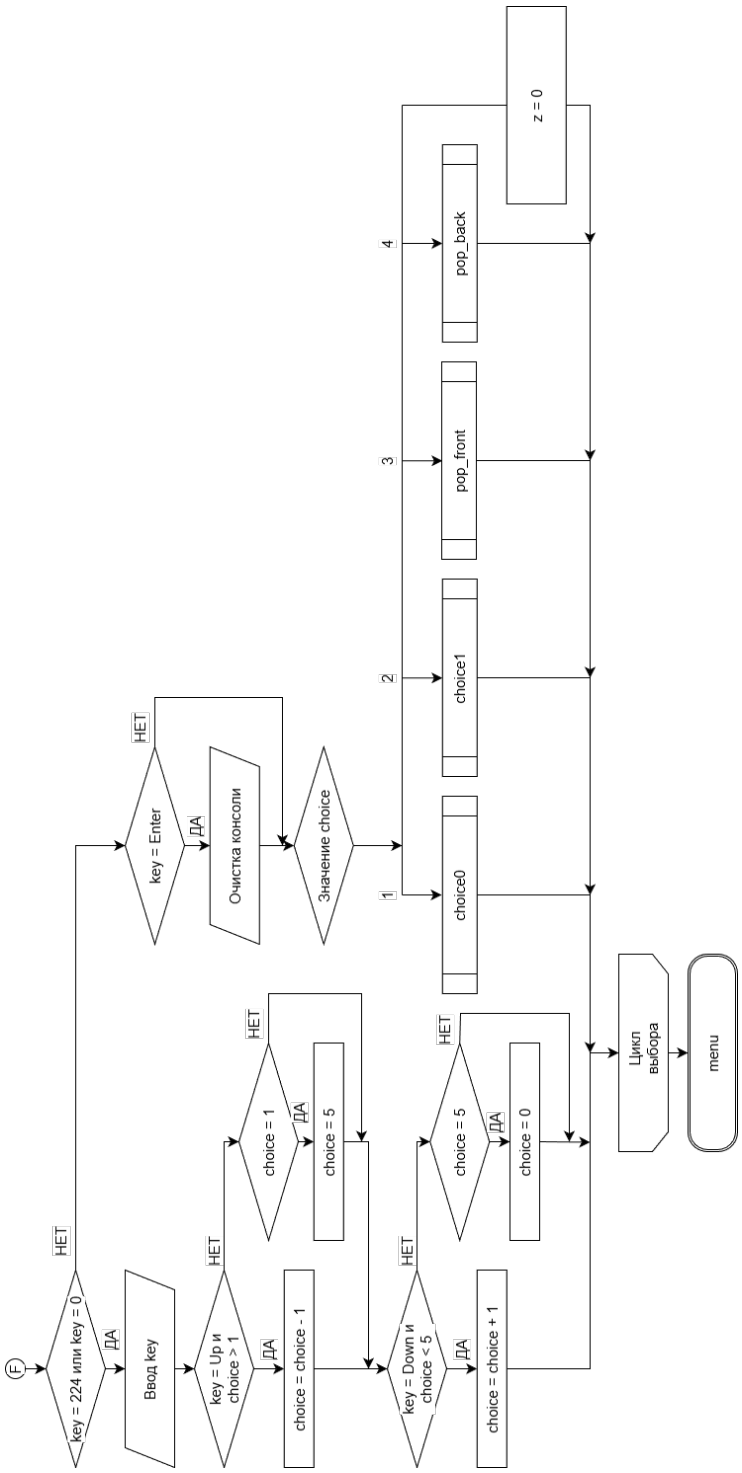

Рис. 2: Вторая часть меню

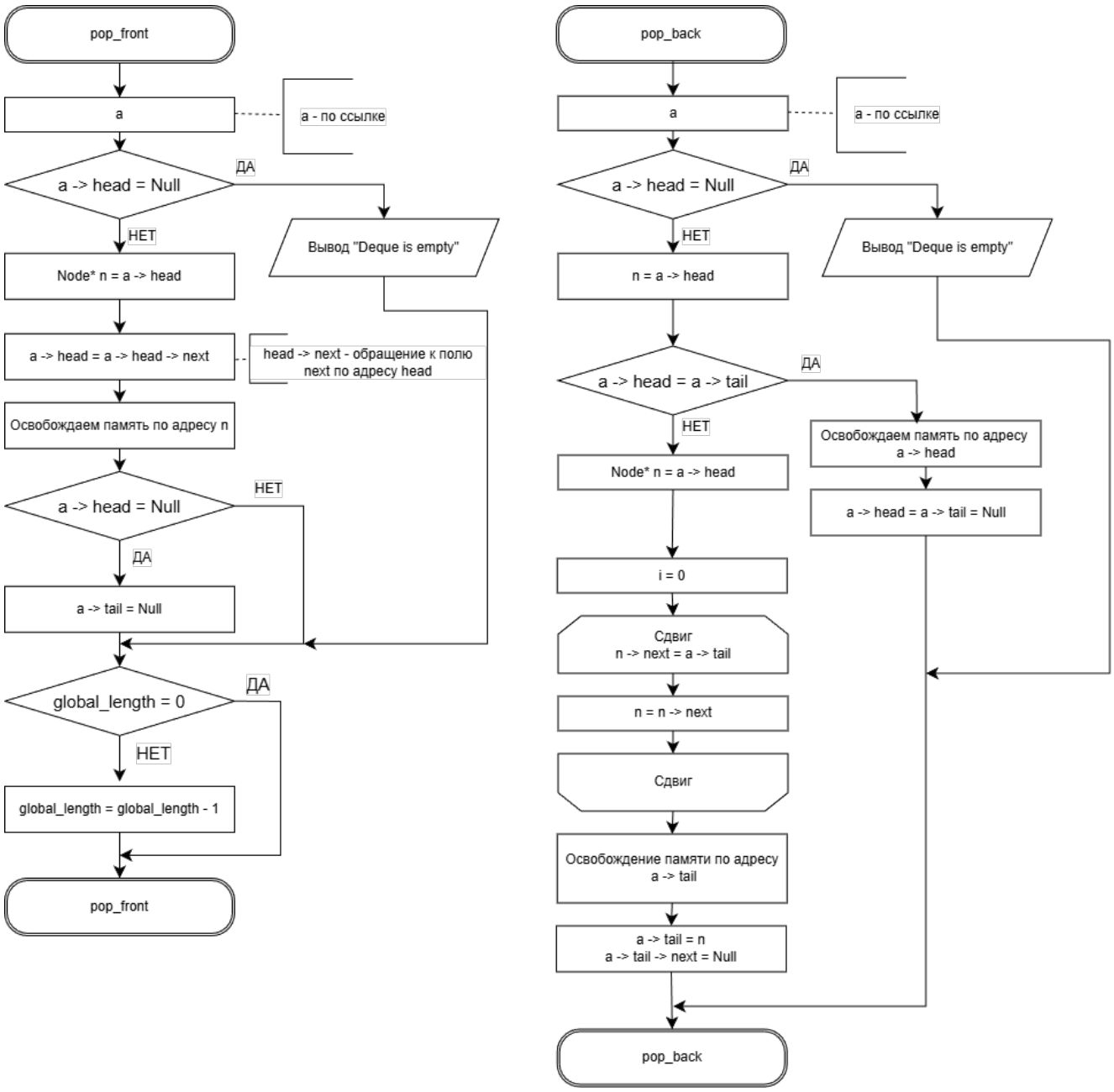

Рис. 3: Исключение элемента сначала и с конца

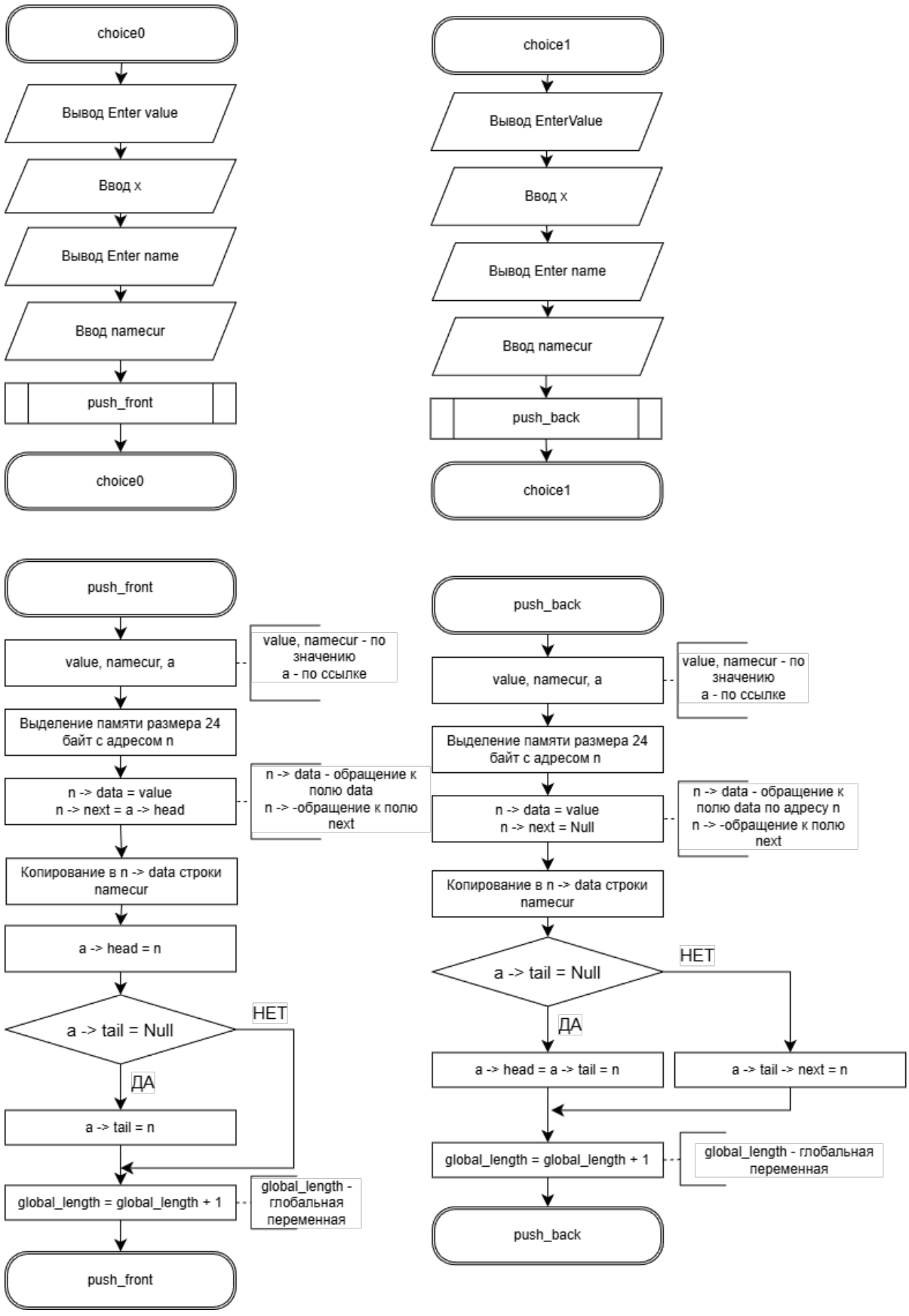

Рис. 4: Добавление сначала и с конца и процедуры выбора пунктов

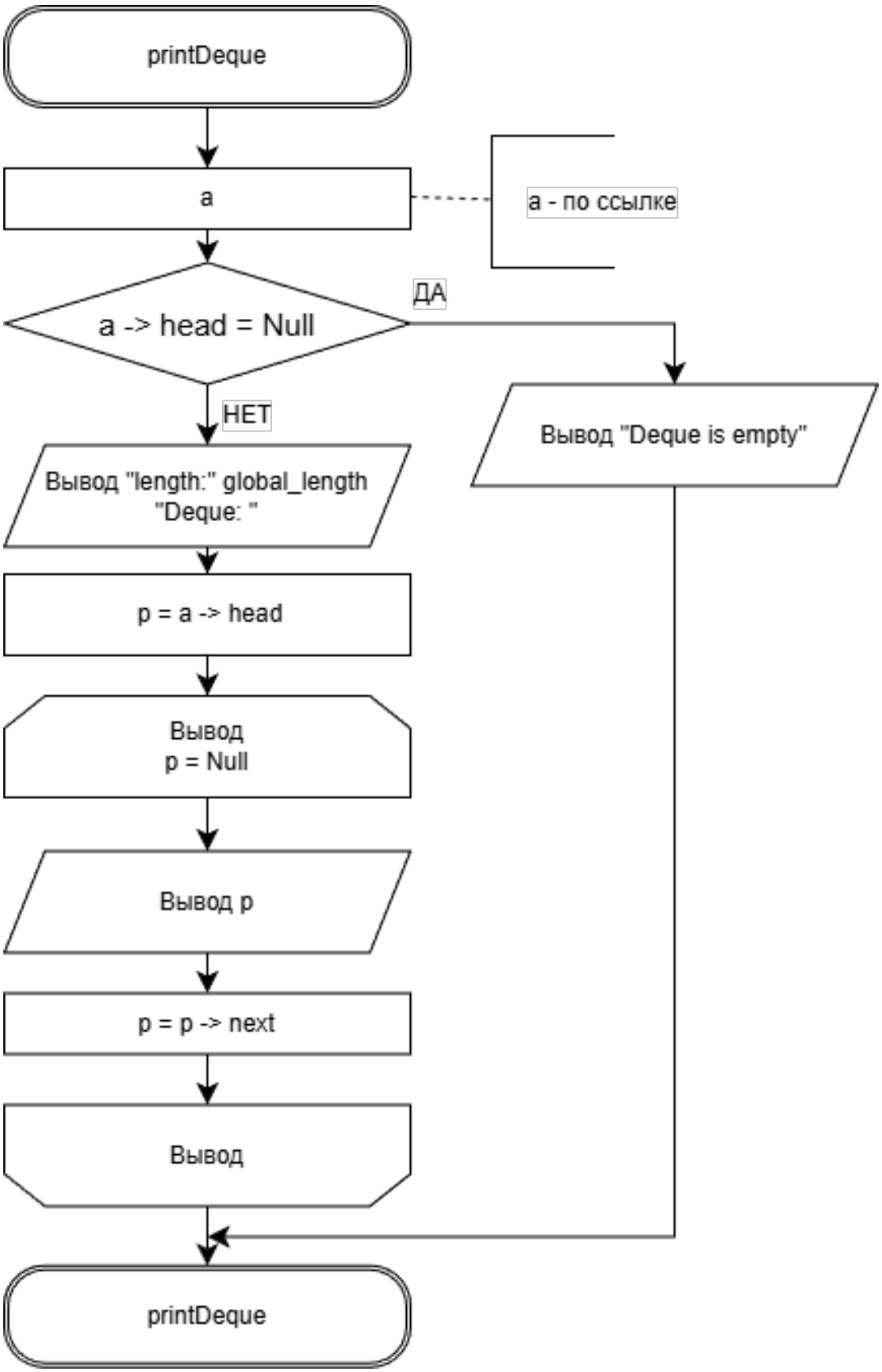

Рис. 5: Вывод дека и его длины

Вывод

Мы разработали консольное приложение полностью повторяющее суть работы дека. Разобрались с базовыми структкруами памяти и научились работать с динамической памятью.