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>

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

typedef struct Node {
    int data;
    struct Node* next;
} Node;
Node* head = NULL;
Node* tail = NULL;

void push_front(int value) {
    Node* n = (Node*)malloc(sizeof(Node));
```

```
    n->data = value;
    n->next = head;
    head = n;
    if (tail == NULL) tail = n;
}

void push_back(int value) {
    Node* n = (Node*)malloc(sizeof(Node));
    n->data = value;
    n->next = NULL;

    if (tail) {
        tail->next = n;
        tail = n;
    }
    else {
        head = tail = n;
    }
}

void choice0() {
    int x;
    printf("Enter value: ");
    while (scanf_s("%d", &x) != 1) {
        printf("Error! Enter a number: ");
        int c;
        while ((c = getchar()) != '\n');
    }
        push_front(x);
}
```

```
void choice1() {
    int x;
    printf("Enter value: ");
    while (scanf_s("%d", &x) != 1) {
        printf("Error! Enter a number: ");
        int c;
        while ((c = getchar()) != '\n');
    }
        push_back(x);

}



void pop_front() {
    if (!head) { printf("Deque is empty\n"); return; }
    Node* n = head;
    head = head->next;
    free(n);
    if (!head)
        tail = NULL;
}

void pop_back() {
    if (!head) { printf("Deque is empty\n"); return; }

    if (head == tail) {
        free(head);
        head = tail = NULL;
```

```
        return;
    }

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

    free(tail);
    tail = n;
    tail->next = NULL;
}

void printDeque() {
    if (!head) { printf("Deque is empty\n"); return; }

    Node* p = head;
    while (p) {
        printf("%d ", p->data);
        p = p->next;
    }
    printf("\n");
}



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

```
};

int choice = 0;
int key;

while (1) {
    system("cls");
    printf("=== DEQUE ===\n\n");
    printDeque();
    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 == 0) {
        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) {
            choice0();
```

```
            }
            else if (choice == 1) {
                choice1();
            }
            else if (choice == 2) pop_front();
            else if (choice == 3) pop_back();
            else if (choice == 4) { return; }
        }
    }
}

int main() {
    menu();
    return 0;
}
```

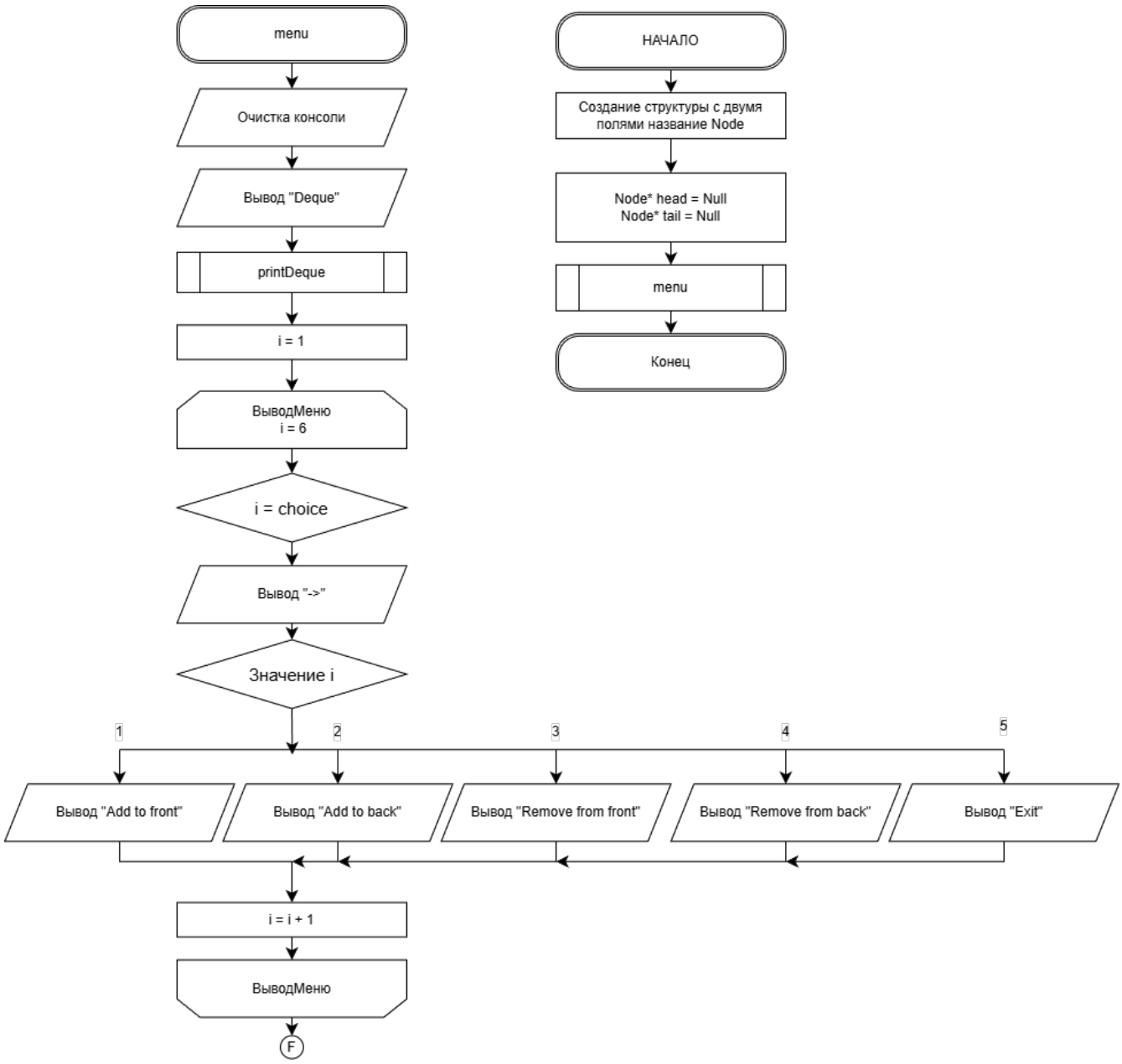

Рис. 1: Первая часть меню и основная программа

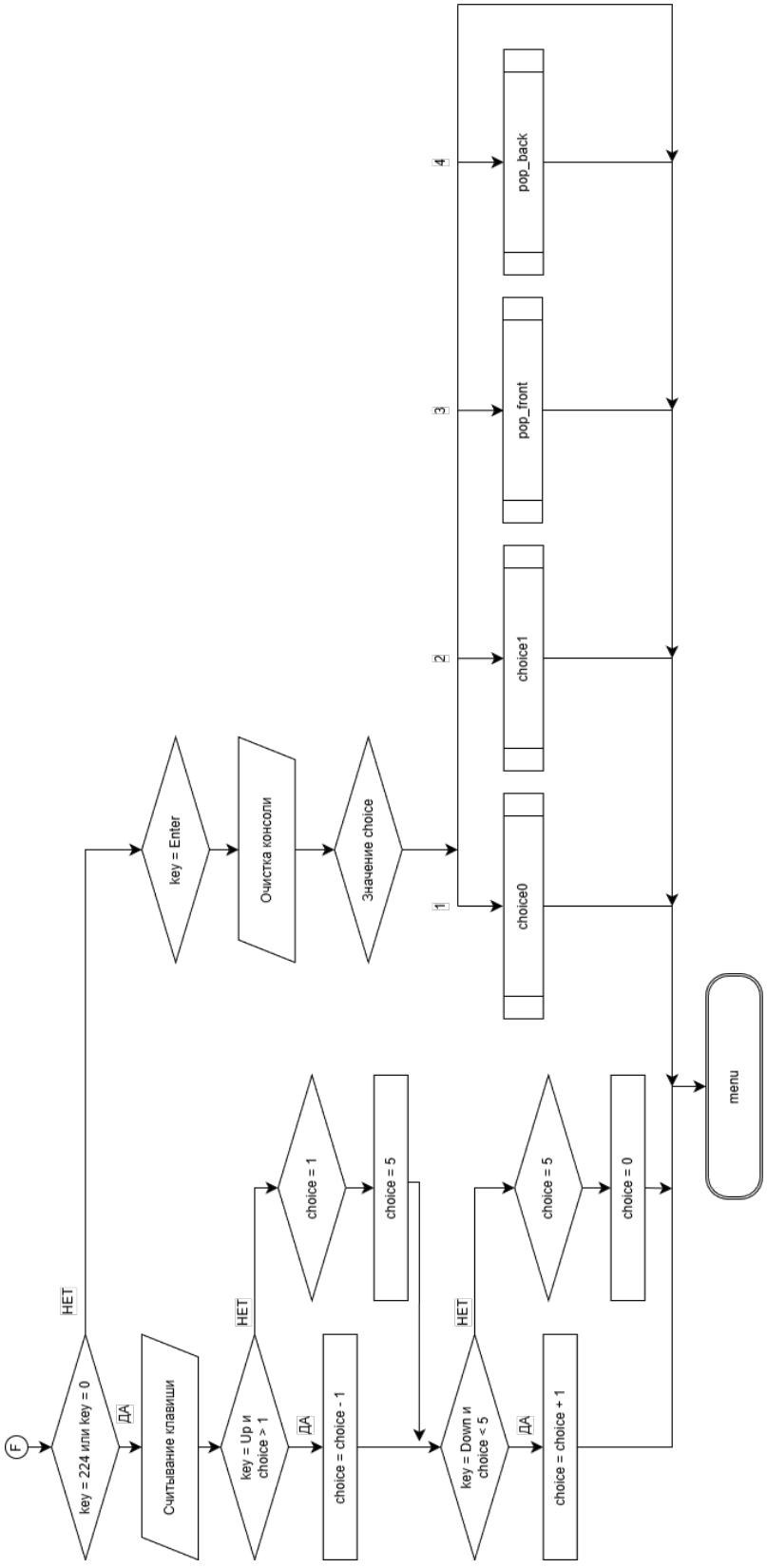

Рис. 2: Вторая часть меню

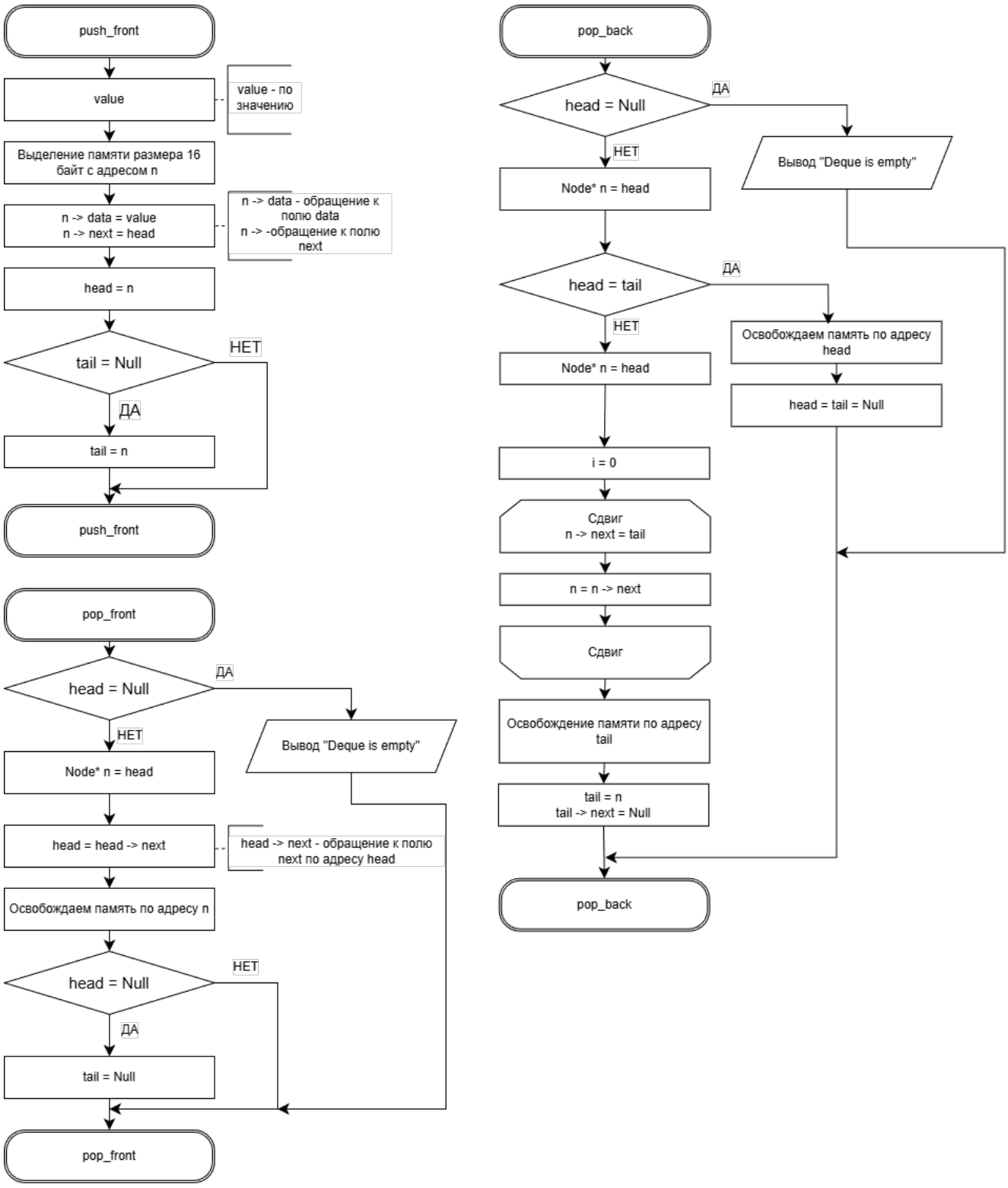

Рис. 3: Добавление и исключение элемента спереди и спереди, исклчение сзади

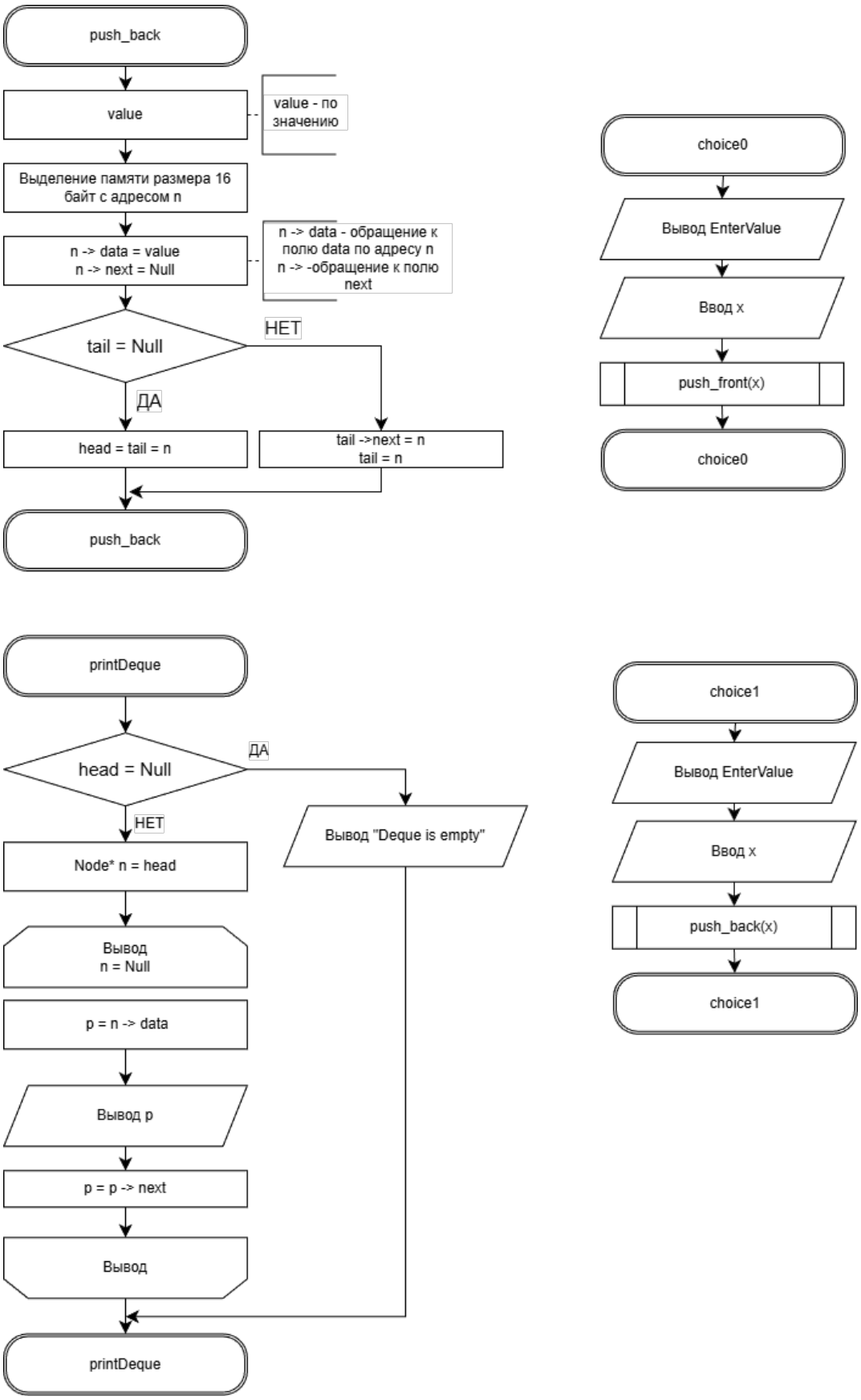

Рис. 4: Добавление сзади, вывод дека и вызов вводов

Вывод

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