



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА

«Динамические структуры данных: списки»

ДИСЦИПЛИНА: «Высокоуровневое программирование»

Выполнил: студент гр. ИУК4-12Б _____ (Карельский М.К.)
(Подпись) (Ф.И.О.)

Проверил: _____ (Пчелинцева Н.И.)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2020

Цель: приобретение практических навыков процедурного программирования средствами языка C++.

Задачи:

1. Познакомиться с видами и отличительными особенностями динамических структур данных.
2. Изучить тип данных указатель и операции над ними.
3. Научиться пользоваться ссылками и операцией взятия адреса переменных.
4. Изучить алгоритмы создания и обработки динамических структур данных

Вариант 2

Задача

В соответствии с индивидуальным заданием необходимо создать программу для обслуживания односвязного списка. Указанные в задании действия должны быть оформлены в виде отдельных функций. Память под очередной элемент динамической структуры данных следует выделять динамически.

Выполнение действий по обслуживанию динамической структуры данных должно производиться в режиме диалога с пользователем. Для каждого из вариантов задания необходимо разработать следующие функции:

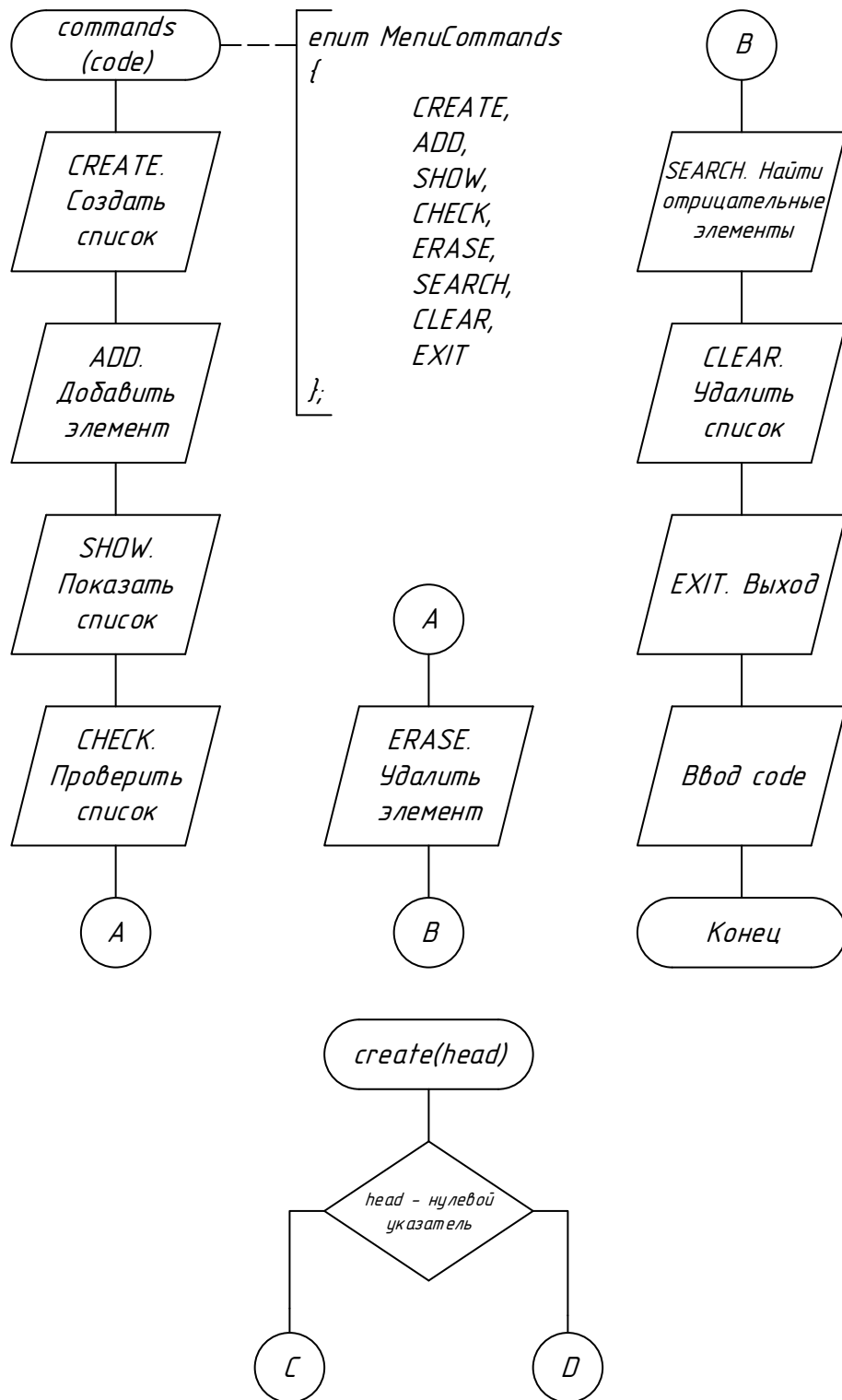
1. Создание списка.
2. Добавление элемента в список:
 - a. в начало списка;
 - b. в конец списка;
 - c. после элемента с заданным номером;
 - d. после элемента с заданным ключом.
3. Вывод содержимого списка на экран.
4. Проверка пустоты списка.
5. Удаление элемента из списка:
 - a. из начала списка;
 - b. из конца списка;
 - c. элемента с заданным номером; d. элемента с заданным ключом.
6. Поиск элементов в списке по выбранному признаку.
7. Удаление списка.

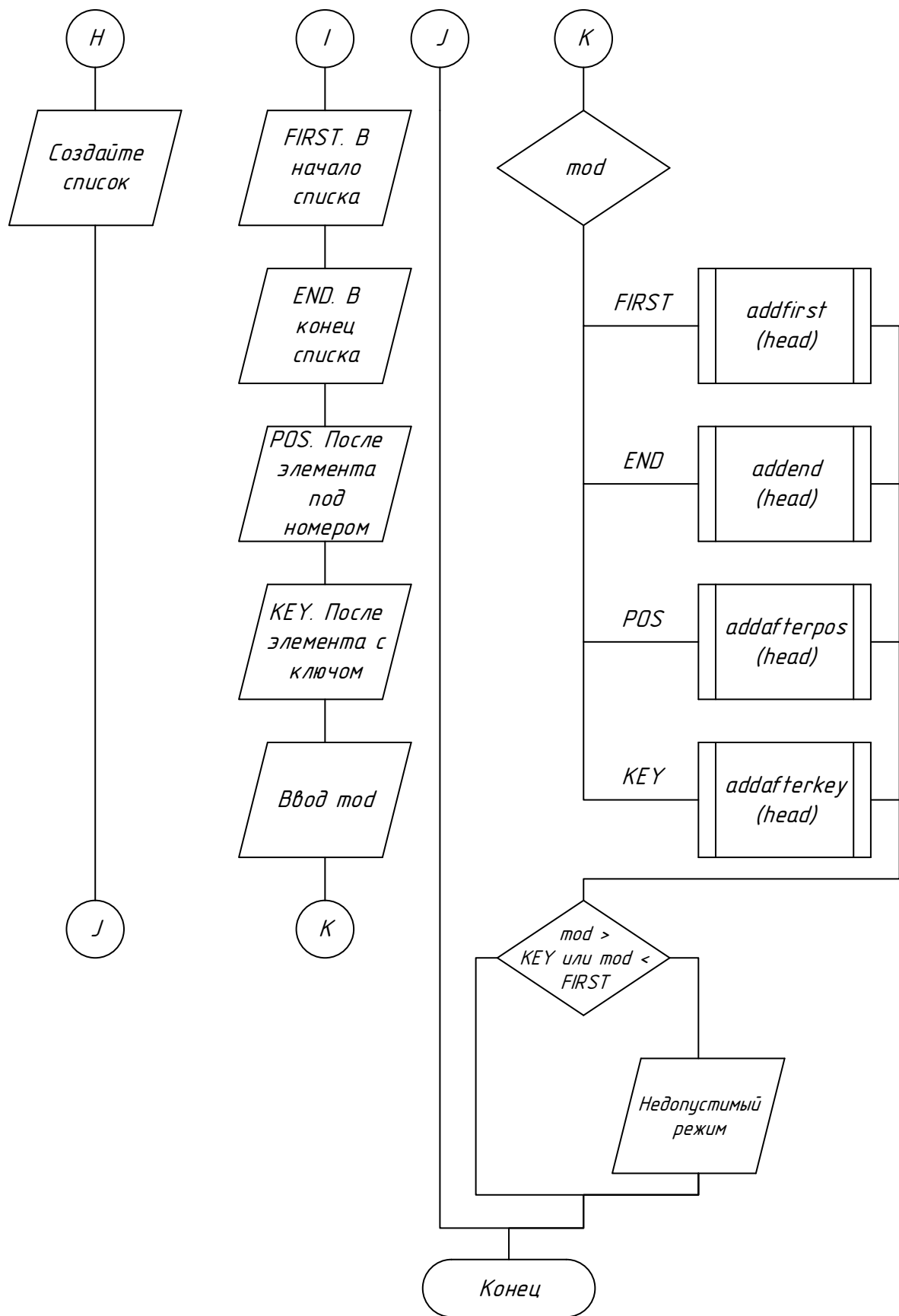
Тип данных списка – вещественные числа.

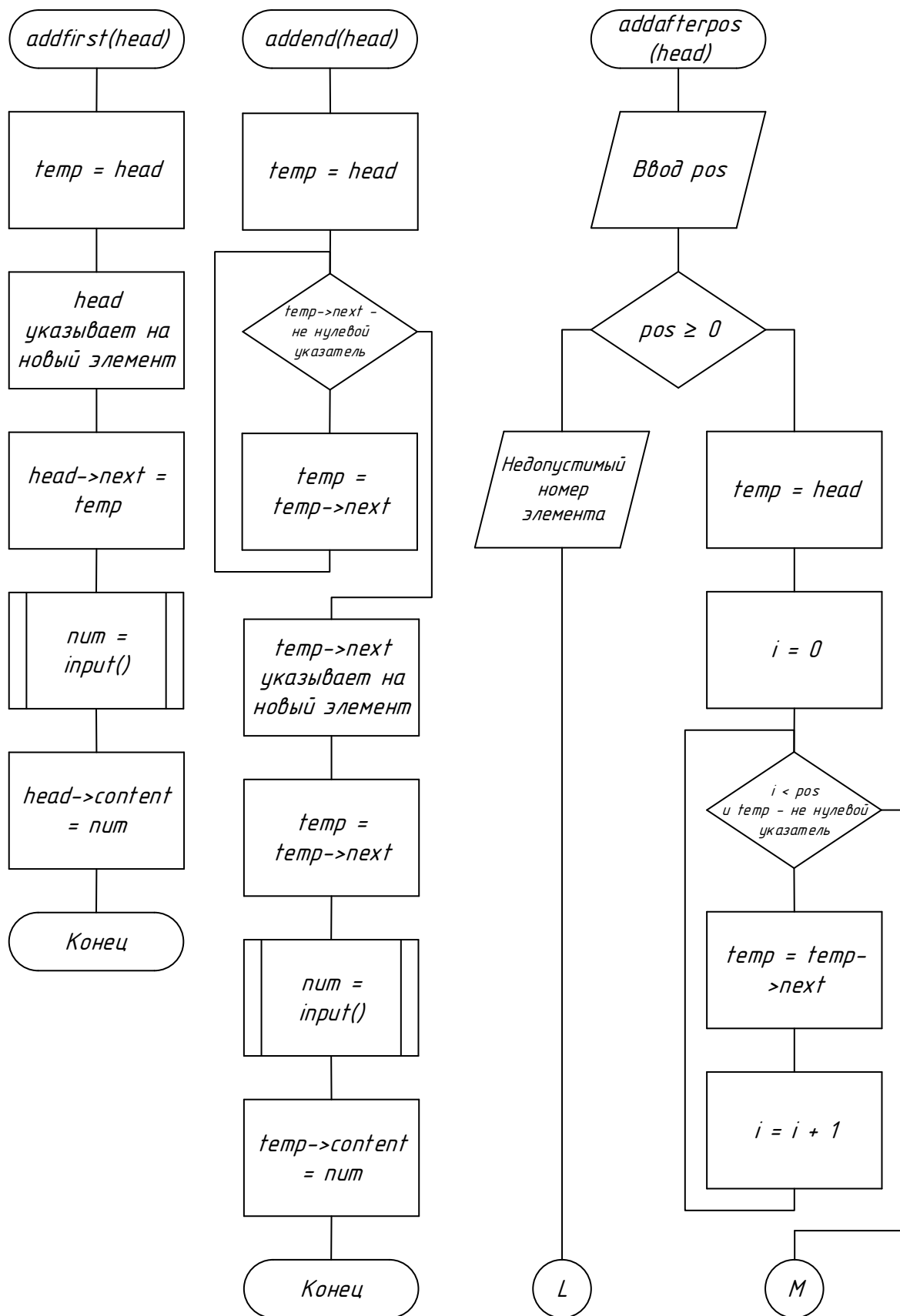
Диапазон чисел – [-19,9; 59,9].

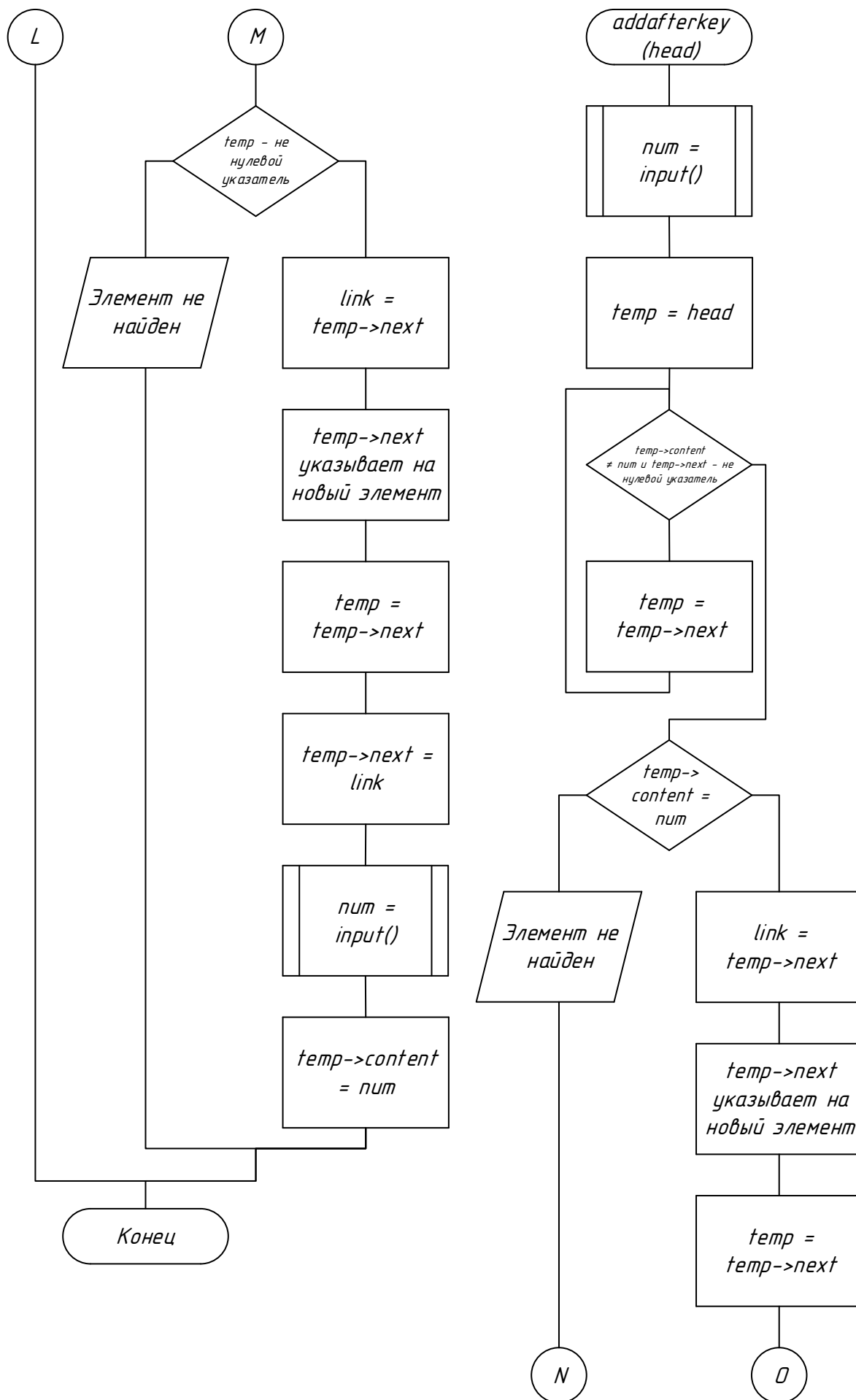
Признак для поиска – отрицательный элемент.

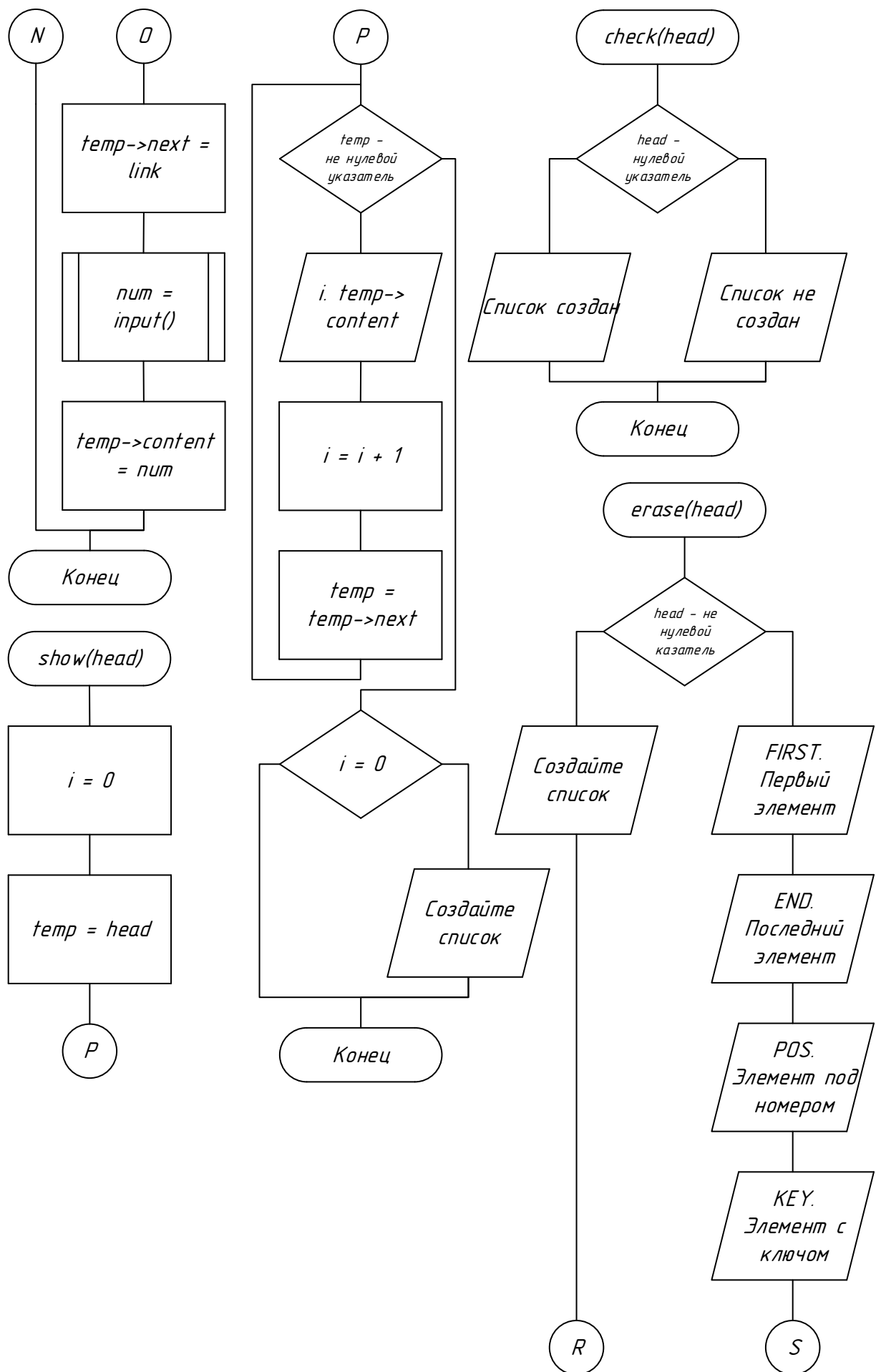
Блок-схема:

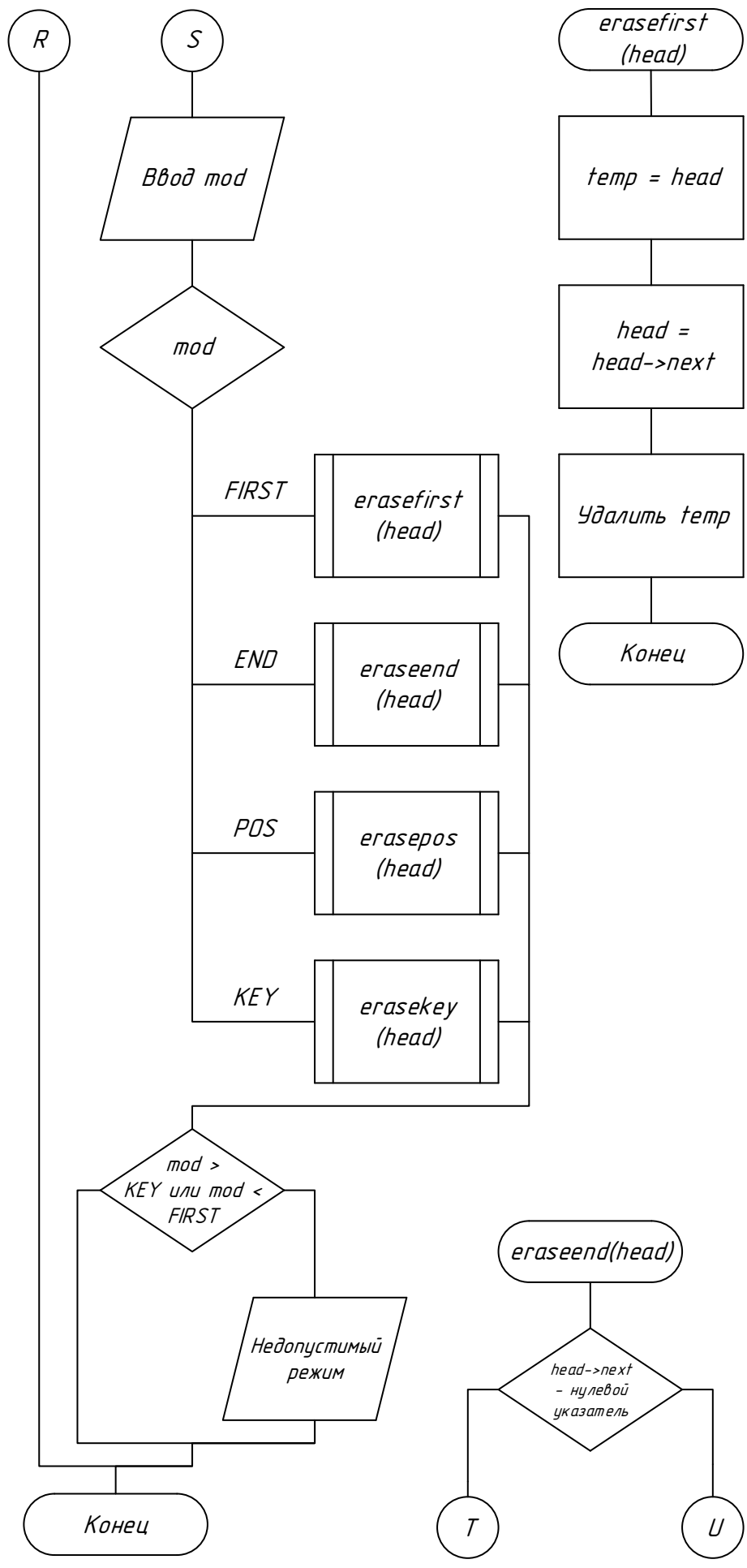


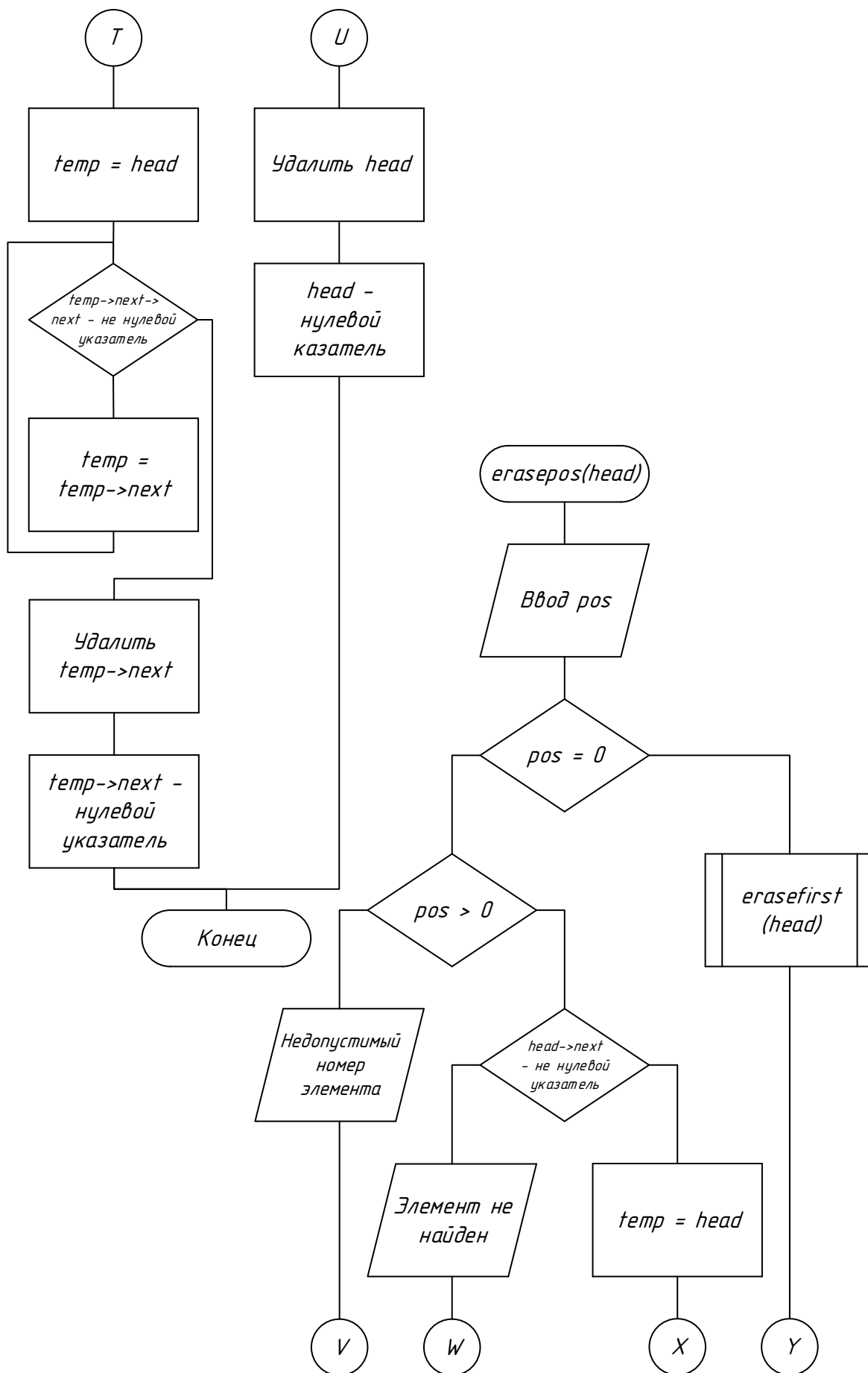


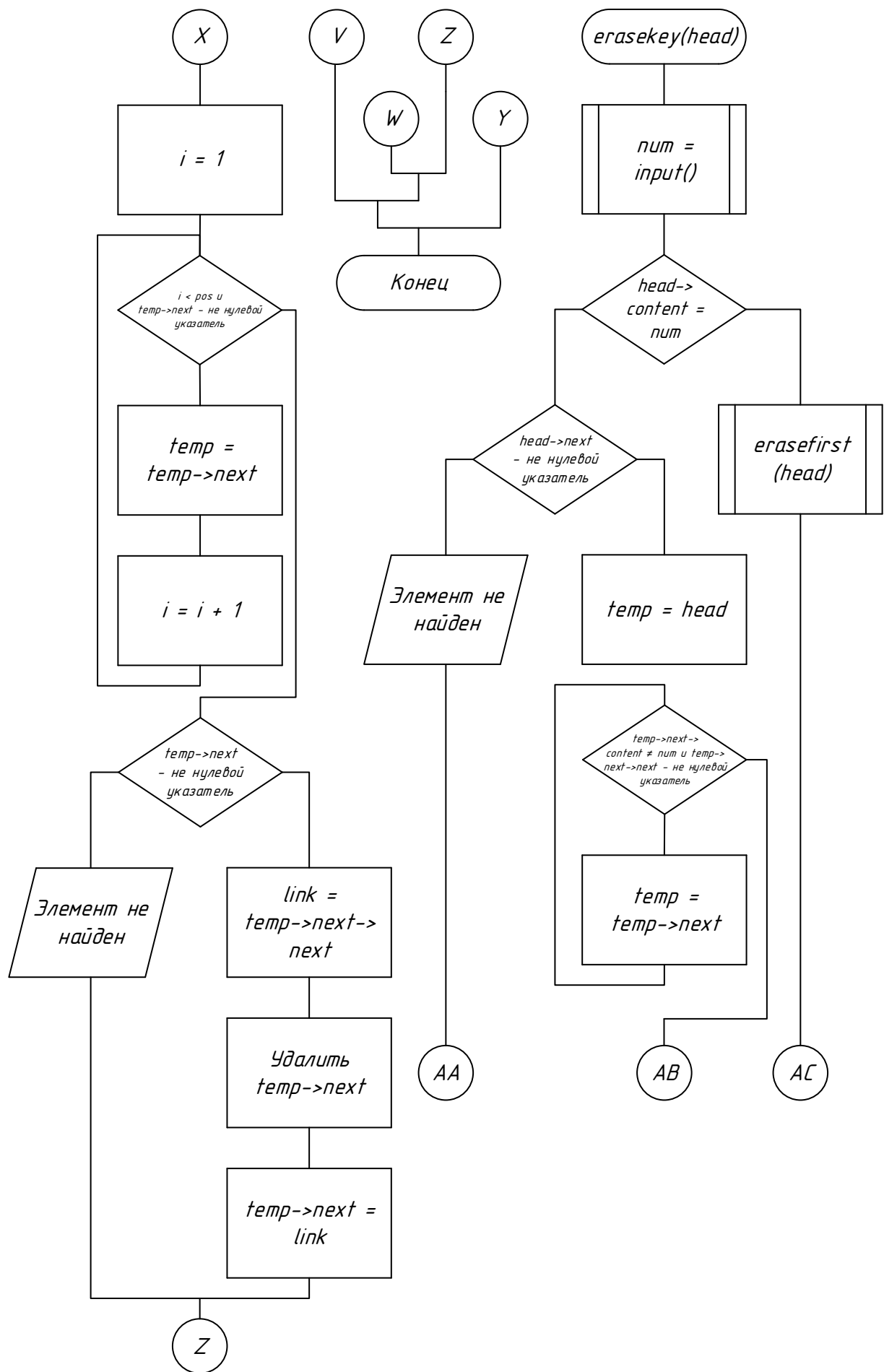


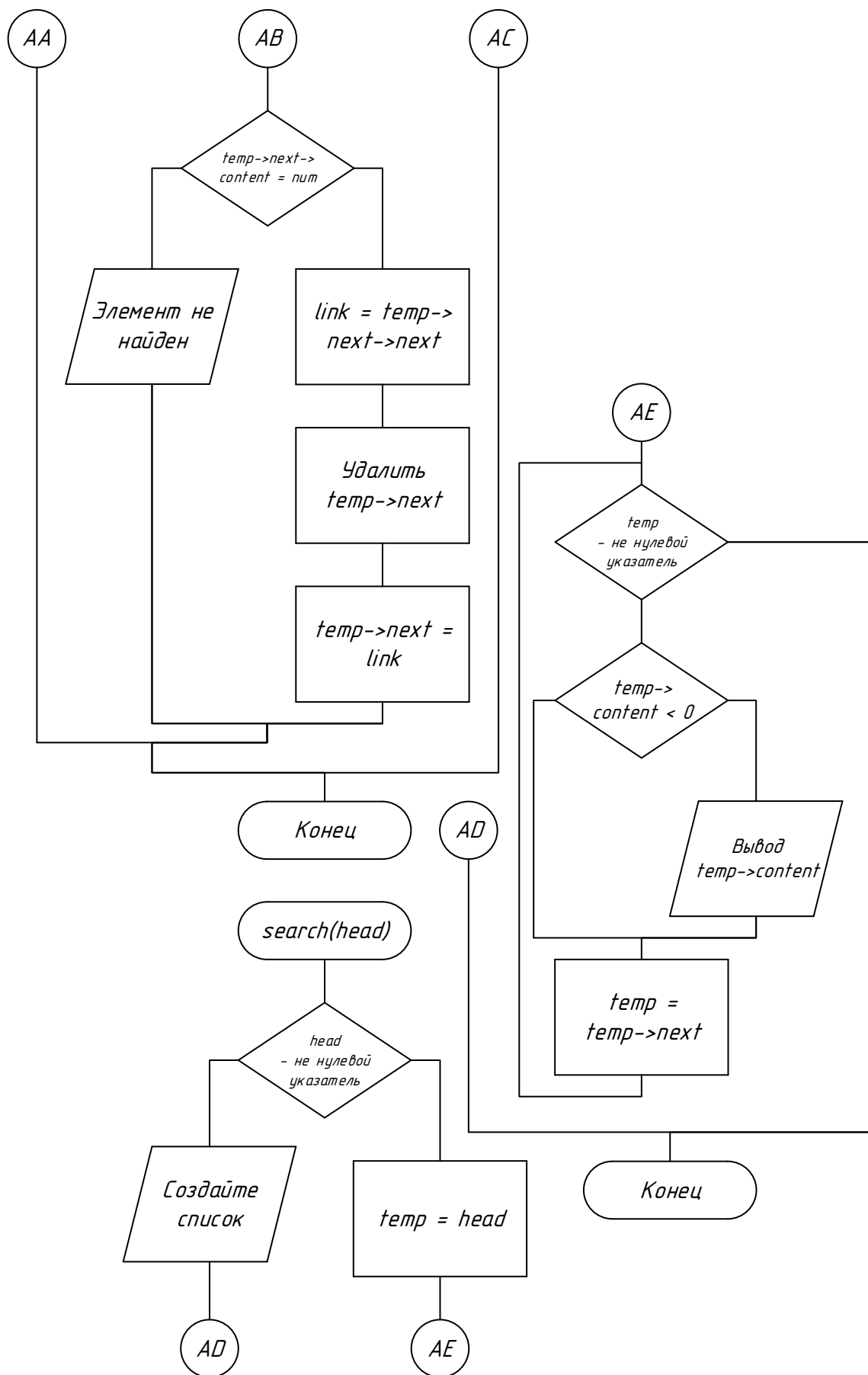


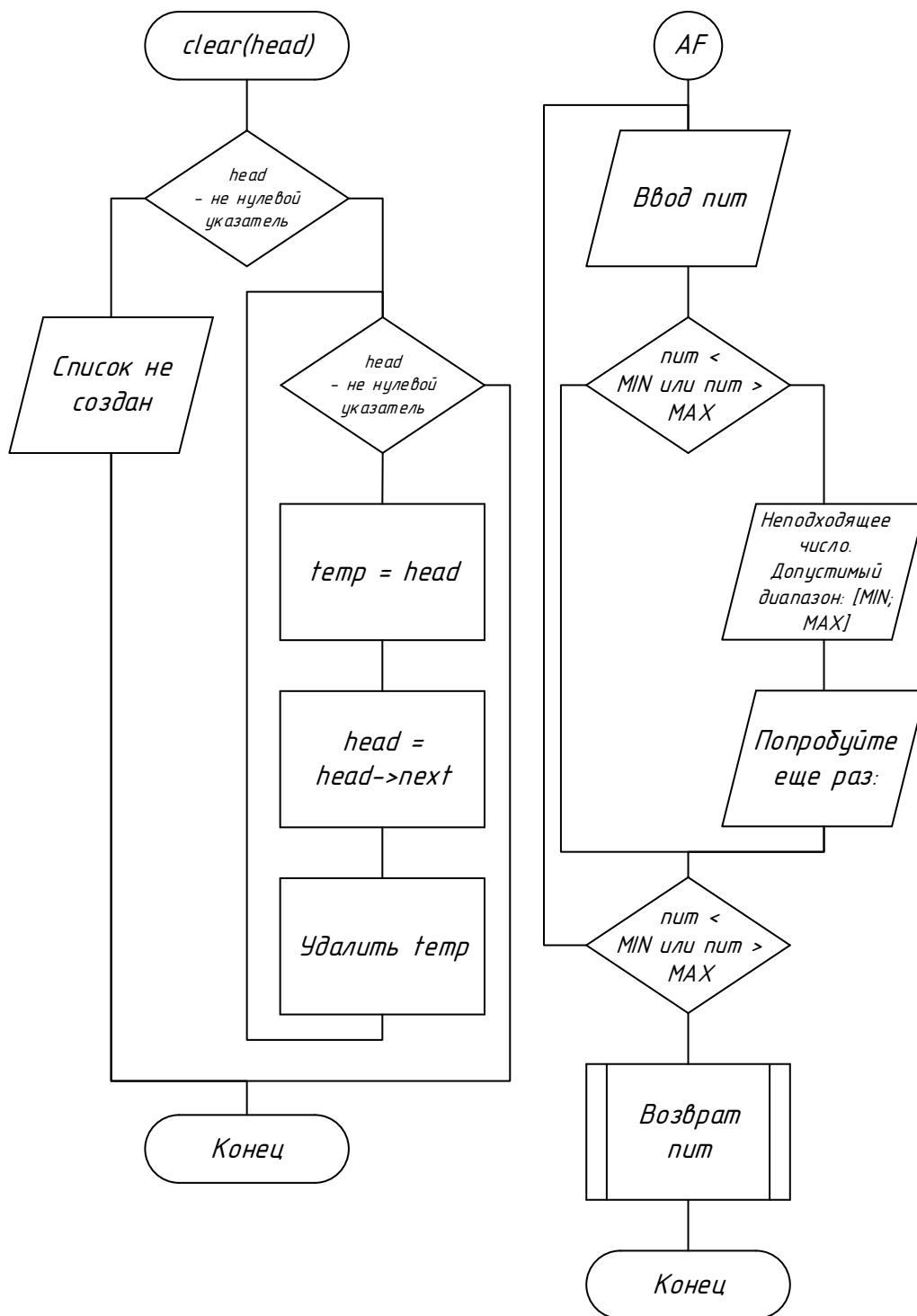




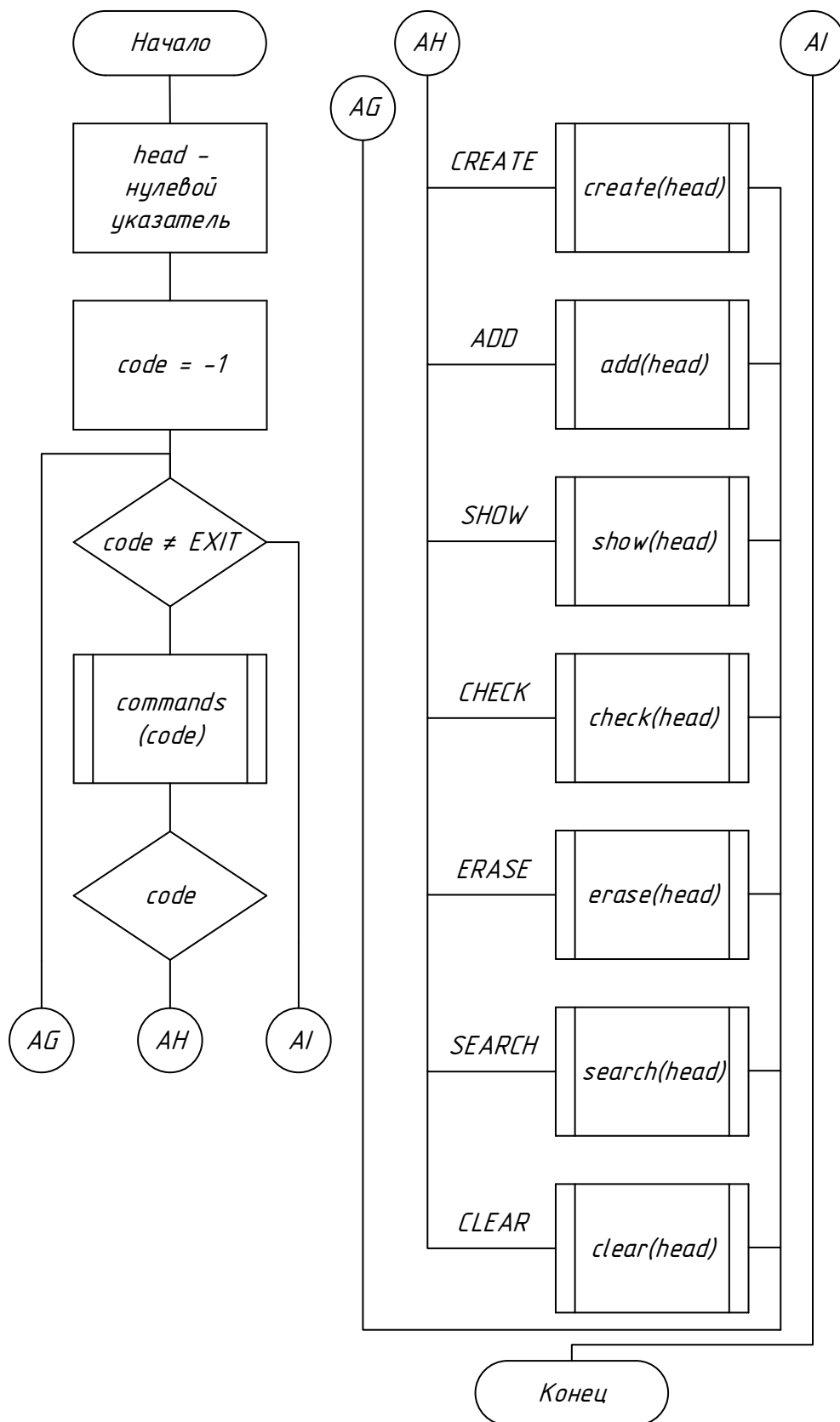








input() — `const float MIN = -19.9;`
 `const float MAX = 59.9;`
 AF



Листинг:

```
#include <iostream>
using namespace std;
```

```

const float MIN = -19.9;
const float MAX = 59.9;

enum MenuCommands
{
    CREATE,
    ADD,
    SHOW,
    CHECK,
    ERASE,
    SEARCH,
    CLEAR,
    EXIT
};

enum Mods
{
    FIRST,
    END,
    POS,
    KEY
};

struct element
{
    element* next = nullptr;
    float content{};
};

void commands(int& code);
void create(element*& head);
void add(element* &head);
void addfirst(element* &head);
void addend(element* head);
void addafterpos(element* head);
void addafterkey(element* head);
void show(element* head);
void check(element* head);
void erase(element*& head);
void erasefirst(element*& head);
void eraseend(element*& head);
void erasepos(element*& head);
void erasekey(element*& head);
void search(element* head);
void clear(element*& head);
float input();

int main()
{
    setlocale(LC_ALL, "Russian");
    element* head = nullptr;

    int code = -1;
    while (code != EXIT)
    {
        commands(code);
        switch (code)
        {
            case CREATE:
                system("cls");
                create(head);
                cout << endl;

```

```

        break;
    case ADD:
        system("cls");
        add(head);
        cout << endl;
        break;
    case SHOW:
        system("cls");
        show(head);
        cout << endl;
        break;
    case CHECK:
        system("cls");
        check(head);
        cout << endl;
        break;
    case ERASE:
        system("cls");
        erase(head);
        cout << endl;
        break;
    case SEARCH:
        system("cls");
        search(head);
        cout << endl;
        break;
    case CLEAR:
        system("cls");
        clear(head);
        break;
    }
}

return 0;
}

void commands(int &code)
{
    cout << "    " << CREATE << ". Создать список" << endl;
    cout << "    " << ADD << ". Добавить элемент" << endl;
    cout << "    " << SHOW << ". Показать список" << endl;
    cout << "    " << CHECK << ". Проверить список" << endl;
    cout << "    " << ERASE << ". Удалить элемент" << endl;
    cout << "    " << SEARCH << ". Найти отрицательные элементы" << endl;
    cout << "    " << CLEAR << ". Удалить список" << endl;
    cout << "    " << EXIT << ". Выход" << endl;
    cout << "Введите код: ";
    cin >> code;
}

void create(element* &head)
{
    if (head == nullptr)
    {
        cout << "Введите кол-во элементов: ";
        int len{};
        cin >> len;

        if (len > 0)
        {
            head = new element{};
            element* temp = head;
            cout << "Введите 0-й элемент: ";

```



```

        float num{};
        num = input();
        temp->content = num;

        for (int i = 1; i < len; i++)
        {
            temp->next = new element{};
            temp = temp->next;
            cout << "Введите " << i << "-й элемент: ";
            num = input();
            temp->content = num;
        }
    }
    else
    {
        cout << "Недопустимое число элементов" << endl;
    }
}
else
{
    cout << "Список уже создан" << endl;
}
}

void add(element* &head)
{
    if (head != nullptr)
    {
        cout << "Режимы" << endl;
        cout << FIRST << ". В начало списка" << endl;
        cout << END << ". В конец списка" << endl;
        cout << POS << ". После элемента под номером" << endl;
        cout << KEY << ". После элемента с ключом" << endl;
        cout << "Выберите режим: ";
        int mod{};
        cin >> mod;
        switch (mod)
        {
            case FIRST:
                addfirst(head);
                break;
            case END:
                addend(head);
                break;
            case POS:
                addafterpos(head);
                break;
            case KEY:
                addafterkey(head);
                break;
        }
        if (mod > KEY || mod < FIRST)
        {
            cout << "Недопустимый режим" << endl;
        }
    }
    else
    {
        cout << "Создайте список" << endl;
    }
}

void addfirst(element* &head)

```

```

{
    element* temp = head;
    head = new element{};
    head->next = temp;
    cout << "Введите элемент: ";
    float num{};
    num = input();
    head->content = num;
}

void addend(element* head)
{
    element* temp = head;
    while (temp->next != nullptr)
    {
        temp = temp->next;
    }
    temp->next = new element{};
    temp = temp->next;
    cout << "Введите элемент: ";
    float num{};
    num = input();
    temp->content = num;
}

void addafterpos(element* head)
{
    cout << "Введите номер элемента: ";
    int pos{};
    cin >> pos;
    if (pos >= 0)
    {
        element* temp = head;
        int i = 0;
        while (i < pos && temp != nullptr)
        {
            temp = temp->next;
            i++;
        }
        if (temp != nullptr)
        {
            element* link = temp->next;
            temp->next = new element{};
            temp = temp->next;
            temp->next = link;
            cout << "Введите элемент: ";
            float num{};
            num = input();
            temp->content = num;
        }
        else
        {
            cout << "Элемент не найден" << endl;
        }
    }
    else
    {
        cout << "Недопустимый номер элемента" << endl;
    }
}

void addafterkey(element* head)
{

```

```

cout << "Введите ключ: ";
float num{};
num = input();
element* temp = head;
while (temp->content != num && temp->next != nullptr)
{
    temp = temp->next;
}
if (temp->content == num)
{
    element* link = temp->next;
    temp->next = new element{};
    temp = temp->next;
    temp->next = link;
    cout << "Введите элемент: ";
    num = input();
    temp->content = num;
}
else
{
    cout << "Элемент не найден" << endl;
}
}

void show(element* head)
{
    int i = 0;
    element* temp = head;
    while (temp != nullptr)
    {
        cout << i << ". " << temp->content << endl;
        i++;
        temp = temp->next;
    }
    if (i == 0)
    {
        cout << "Создайте список" << endl;
    }
}

void check(element* head)
{
    if (head == nullptr)
    {
        cout << "Список не создан" << endl;
    }
    else
    {
        cout << "Список создан" << endl;
    }
}

void erase(element* &head)
{
    if (head != nullptr)
    {
        cout << "Режимы" << endl;
        cout << FIRST << ". Первый элемент" << endl;
        cout << END << ". Последний элемент" << endl;
        cout << POS << ". Элемент под номером" << endl;
        cout << KEY << ". Элемент с ключом" << endl;
        cout << "Выберите режим: ";
        int mod{};
    }
}

```

```

        cin >> mod;
        switch (mod)
        {
            case FIRST:
                erasefirst(head);
                break;
            case END:
                eraseend(head);
                break;
            case POS:
                erasepos(head);
                break;
            case KEY:
                erasekey(head);
                break;
        }
        if (mod > KEY || mod < FIRST)
        {
            cout << "Недопустимый режим" << endl;
        }
    }
    else
    {
        cout << "Создайте список" << endl;
    }
}

void erasefirst(element* &head)
{
    element* temp = head;
    head = head->next;
    delete temp;
}

void eraseend(element* &head)
{
    if (head->next == nullptr)
    {
        delete head;
        head = nullptr;
    }
    else
    {
        element* temp = head;
        while (temp->next->next != nullptr)
        {
            temp = temp->next;
        }
        delete temp->next;
        temp->next = nullptr;
    }
}

void erasepos(element* &head)
{
    cout << "Введите номер элемента: ";
    int pos{};
    cin >> pos;
    if (pos == 0)
    {
        erasefirst(head);
    }
    else if (pos > 0)

```

```

{
    if (head->next != nullptr)
    {
        element* temp = head;
        int i = 1;
        while (i < pos && temp->next != nullptr)
        {
            temp = temp->next;
            i++;
        }
        if (temp->next != nullptr)
        {
            element* link = temp->next->next;
            delete temp->next;
            temp->next = link;
        }
        else
        {
            cout << "Элемент не найден" << endl;
        }
    }
    else
    {
        cout << "Элемент не найден" << endl;
    }
}

else
{
    cout << "Недопустимый номер элемента" << endl;
}
}

void erasekey(element* &head)
{
    cout << "Введите ключ: ";
    float num{};
    num = input();
    if (head->content == num)
    {
        erasefirst(head);
    }
    else if (head->next != nullptr)
    {
        element* temp = head;
        while (temp->next->content != num && temp->next->next != nullptr)
        {
            temp = temp->next;
        }
        if (temp->next->content == num)
        {
            element* link = temp->next->next;
            delete temp->next;
            temp->next = link;
        }
        else
        {
            cout << "Элемент не найден" << endl;
        }
    }
    else
    {
        cout << "Элемент не найден" << endl;
    }
}

```

```

}

void search(element* head)
{
    if (head != nullptr)
    {
        cout << "Найденные отрицательные элементы:" << endl;
        element* temp = head;
        while (temp != nullptr)
        {
            if (temp->content < 0)
            {
                cout << temp->content << endl;
            }
            temp = temp->next;
        }
    }
    else
    {
        cout << "Создайте список" << endl;
    }
}

void clear(element* &head)
{
    if (head != nullptr)
    {
        while (head != nullptr)
        {
            element* temp = head;
            head = head->next;
            delete temp;
        }
    }
    else
    {
        cout << "Список не создан" << endl << endl;
    }
}

float input()
{
    float num{};
    do
    {
        cin >> num;
        if (num < MIN || num > MAX)
        {
            cout << "Неподходящее число. Допустимый диапазон: [" << MIN
<< "; " << MAX << "]" << endl;
            cout << "Попробуйте еще раз: ";
        }
    } while (num < MIN || num > MAX);
    return num;
}

```

Тестирование:

0. Создать список
1. Добавить элемент

2. Показать список
3. Проверить список
4. Удалить элемент
5. Найти отрицательные элементы
6. Удалить список
7. Выход

Введите код:

При вводе 1, 2, 4 или 5:

Создайте список

0. Создать список
1. Добавить элемент
2. Показать список
3. Проверить список
4. Удалить элемент
5. Найти отрицательные элементы
6. Удалить список
7. Выход

Введите код:

При вводе 3 или 6:

Список не создан

0. Создать список
1. Добавить элемент
2. Показать список
3. Проверить список
4. Удалить элемент
5. Найти отрицательные элементы
6. Удалить список
7. Выход

Введите код:

При вводе 0 и последующем заполнении:

Введите кол-во элементов: 7

Введите 0-й элемент: -1.1

Введите 1-й элемент: 2.2

Введите 2-й элемент: -3.3

Введите 3-й элемент: 4.4

Введите 4-й элемент: -5.5

Введите 5-й элемент: 6.6

Введите 6-й элемент: -7.7

0. Создать список
1. Добавить элемент
2. Показать список
3. Проверить список
4. Удалить элемент
5. Найти отрицательные элементы
6. Удалить список

7. Выход

Введите код:

При вводе 0:

Список уже создан

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 3:

Список создан

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 0

Введите элемент: 100

Неподходящее число. Допустимый диапазон: [-19.9; 59.9]

Попробуйте еще раз: 10

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 1

Введите элемент: -123.45

Неподходящее число. Допустимый диапазон: [-19.9; 59.9]

Попробуйте еще раз: -6.9

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 2

Введите номер элемента: 3

Введите элемент: 25

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 2

Введите номер элемента: 100

Элемент не найден

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 2

Введите номер элемента: -10

Недопустимый номер элемента

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 2

Введите номер элемента: 9

Введите элемент: 32

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 3

Введите ключ: 25

Введите элемент: 20

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 1 и последующем заполнении:

Режимы

- 0. В начало списка
- 1. В конец списка
- 2. После элемента под номером
- 3. После элемента с ключом

Выберите режим: 3

Введите ключ: 11

Элемент не найден

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 2:

- 0. 10
- 1. -1.1
- 2. 2.2
- 3. -3.3
- 4. 25
- 5. 20
- 6. 4.4
- 7. -5.5
- 8. 6.6
- 9. -7.7

10. -6.9

11. 32

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 5:

Найденные отрицательные элементы:

-1.1

-3.3

-5.5

-7.7

-6.9

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 4 и последующем заполнении:

Режимы

0. Первый элемент

1. Последний элемент

2. Элемент под номером

3. Элемент с ключом

Выберите режим: 0

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 4 и последующем заполнении:

Режимы

- 0. Первый элемент
- 1. Последний элемент
- 2. Элемент под номером
- 3. Элемент с ключом

Выберите режим: 1

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 4 и последующем заполнении:

Режимы

- 0. Первый элемент
- 1. Последний элемент
- 2. Элемент под номером
- 3. Элемент с ключом

Выберите режим: 2

Введите номер элемента: 20

Элемент не найден

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 4 и последующем заполнении:

Режимы

- 0. Первый элемент
- 1. Последний элемент
- 2. Элемент под номером
- 3. Элемент с ключом

Выберите режим: 2

Введите номер элемента: -10

Недопустимый номер элемента

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список

7. Выход

Введите код:

При вводе 4 и последующем заполнении:

Режимы

0. Первый элемент

1. Последний элемент

2. Элемент под номером

3. Элемент с ключом

Выберите режим: 2

Введите номер элемента: 5

0. Создать список

1. Добавить элемент

2. Показать список

3. Проверить список

4. Удалить элемент

5. Найти отрицательные элементы

6. Удалить список

7. Выход

Введите код:

При вводе 4 и последующем заполнении:

Режимы

0. Первый элемент

1. Последний элемент

2. Элемент под номером

3. Элемент с ключом

Выберите режим: 3

Введите ключ: 1

Элемент не найден

0. Создать список

1. Добавить элемент

2. Показать список

3. Проверить список

4. Удалить элемент

5. Найти отрицательные элементы

6. Удалить список

7. Выход

Введите код:

При вводе 4 и последующем заполнении:

Режимы

0. Первый элемент

1. Последний элемент

2. Элемент под номером

3. Элемент с ключом

Выберите режим: 3

Введите ключ: 20

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 2:

- 0. -1.1
- 1. 2.2
- 2. -3.3
- 3. 25
- 4. -5.5
- 5. 6.6
- 6. -7.7
- 7. -6.9

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 6:

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список
- 7. Выход

Введите код:

При вводе 3:

Список не создан

- 0. Создать список
- 1. Добавить элемент
- 2. Показать список
- 3. Проверить список
- 4. Удалить элемент
- 5. Найти отрицательные элементы
- 6. Удалить список

7. Выход

Введите код:

Вывод: были получены практические навыки создания динамического односвязанного списка, добавления в него новых элементов, удаления из него выбранных элементов, удаления списка, проверки списка на пустоту, поиска в нем элементов по определенному признаку.