



**«Московский государственный технический университет  
имени Н.Э. Баумана»  
(национальный исследовательский университет)  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**О т ч е т**

**по лабораторной работе № 6**

**Название лабораторной работы: Списковые структуры.**

**Дисциплина: Алгоритмизация и программирование.**

Студент гр. ИУ6-14Б

(Подпись, дата)

**21.09.2024 Д.А.Пасхальная**

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

**21.09.2024 О.А.Веселовская**

(И.О. Фамилия)

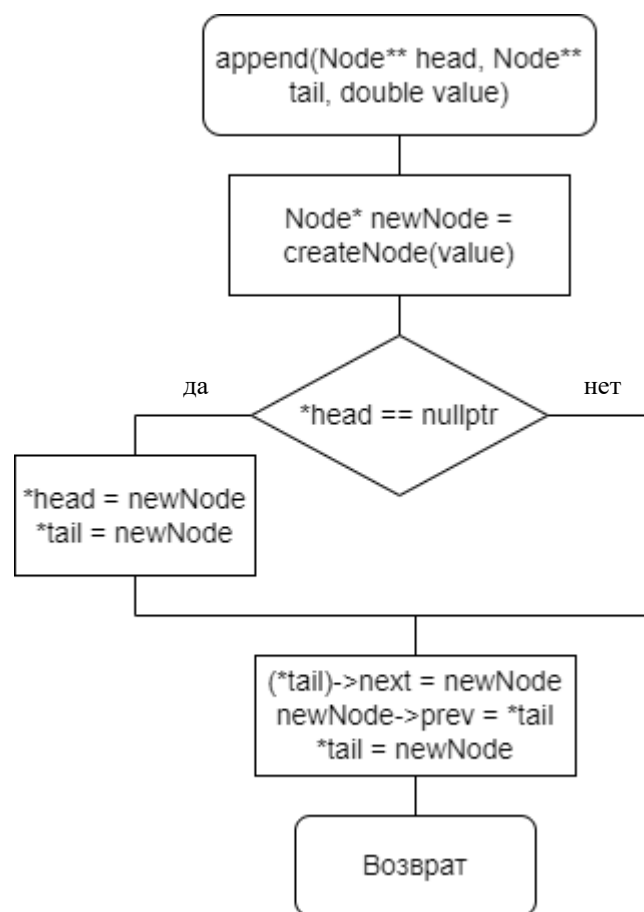
**Цель работы** – Решить поставленную задачу, используя двумерные списки.

**Задание** – С клавиатуры вводится последовательность вещественных чисел  $x_1, x_2, x_3, \dots, x_n$ ,  $n > 2$ . Вывести последовательность  $(x_1 + x_2 + 2x_n)$ ;  $(x_2 + x_3 + 2x_{n-1})$ ;  $(x_3 + x_4 + 2x_{n-2})$  ...  $(x_{n-1} + x_n + 2x_2)$ . Использовать двусвязный список.

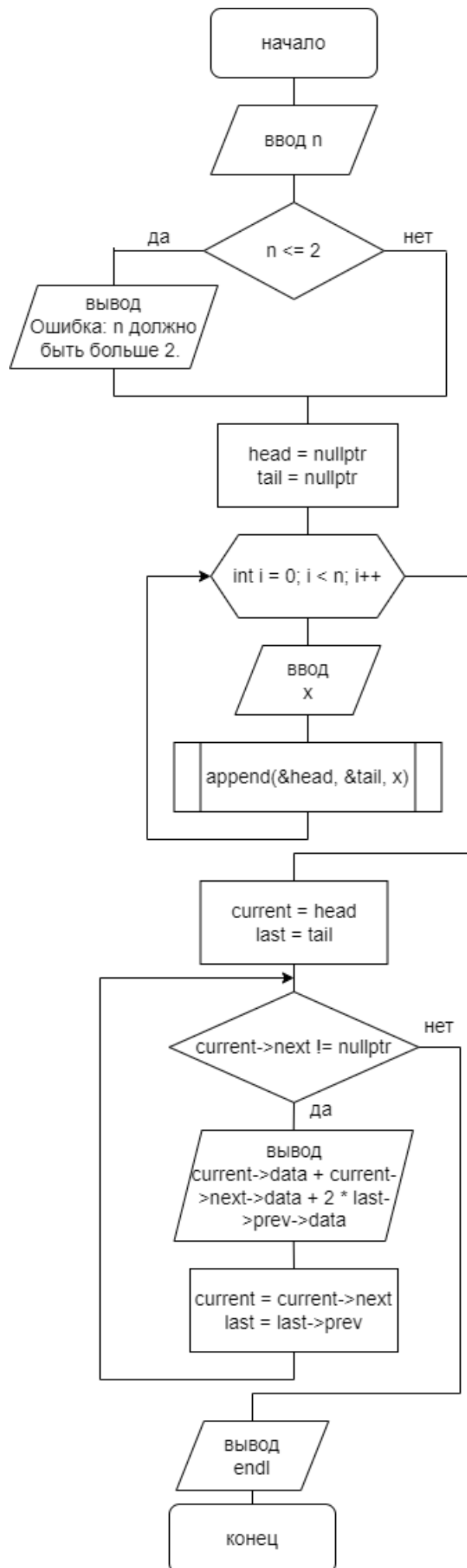
**Ход работы:**

- Изображение схемы алгоритмов программы.
- Написание программы.
- Тестирование программы при различных тестовых данных.
- Вывод.

Для начала составим схему алгоритма программы:



**Рисунок 1** – Схема алгоритма используемой подпрограммы



## Рисунок 2 – Схема алгоритма основной программы

Далее напишем код программы:

```
1  #include <iostream>
2
3  using namespace std;
4
5  struct Node {
6      double data;
7      Node* prev;
8      Node* next;
9  };
10
11 Node* createNode(double value) {
12     Node* newNode = new Node;
13     newNode->data = value;
14     newNode->prev = nullptr;
15     newNode->next = nullptr;
16     return newNode;
17 }
18
19 void append(Node** head, Node** tail, double value) {
20     Node* newNode = createNode(value);
21     if (*head == nullptr) {
22         *head = newNode;
23         *tail = newNode;
24         return;
25     }
26     (*tail)->next = newNode;
27     newNode->prev = *tail;
28     *tail = newNode;
29 }
30
31 void printList(Node* head) {
32     while (head != nullptr) {
33         cout << head->data << " ";
34         head = head->next;
35     }
36     cout << endl;
37 }
38
39 int main() {
40     setlocale(LC_ALL, "Russian");
41
42     int n;
43     double x;
44
45     cout << "Введите количество чисел (n > 2): ";
46     cin >> n;
47
48     if (n <= 2) {
49         cout << "Ошибка: n должно быть больше 2." << endl;
50         return 1;
51     }
52
53     Node* head = nullptr;
54     Node* tail = nullptr;
```

```

56     cout << "Введите числа:\n";
57     for (int i = 0; i < n; i++) {
58         cin >> x;
59         append(&head, &tail, x);
60     }
61
62     Node* current = head;
63     Node* last = tail;
64
65     while (current->next != nullptr) {
66         cout << (current->data + current->next->data + 2 * last->prev->data) << " ";
67         current = current->next;
68         last = last->prev;
69     }
70
71     cout << endl;
72
73     return 0;
74 }
75

```

**Рисунок 3** – Код программы

Далее введём несколько тестирующих команд, проверяющих функционал написанной программы:

```

Введите количество чисел (n > 2): 5
Введите числа:
1 6 4 8 9
23 18 24 19

```

**Рисунок 4** – Тестовые команды часть 1

```

Введите количество чисел (n > 2): 8
Введите числа:
2 4 6 8 0 2 3 1
12 14 14 24 14 13 8

```

**Рисунок 5** – Тестовые команды часть 2

```

Введите количество чисел (n > 2): 12
Введите числа:
1 2 6 8 9 5 3 2 6 5 10 13
23 18 26 21 20 18 23 24 23 19 25

```

**Рисунок 6** – Тестовые команды часть 3

Как видно из рисунков, программа работает корректно на введенных тестовых данных.

**Вывод:** в ходе лабораторной работы были получены навыки работы с двусвязным списком.