

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

по лабораторной работе № 6

Дисциплина: Объектно-ориентированное программирование

А.М.Минитаева
(И.О. Фамилия)

Цель работы

Научиться работать с динамическими структурами данных.

Задание

С клавиатуры вводится символьная строка, содержащая последовательность вещественных чисел $x_1, x_2, x_3 \dots x_n$ ($n > 2$) в символьном представлении. Написать программу, формирующую из этих чисел двунаправленный список, предварительно преобразовав числа в вещественную форму представления. Вычислить $x_1 * x_n; x_2 * x_{n-1}; x_3 * x_{n-2} \dots x_n * x_1$ и составить из произведений новый список. Вывести на печать оба списка.

Проект программы

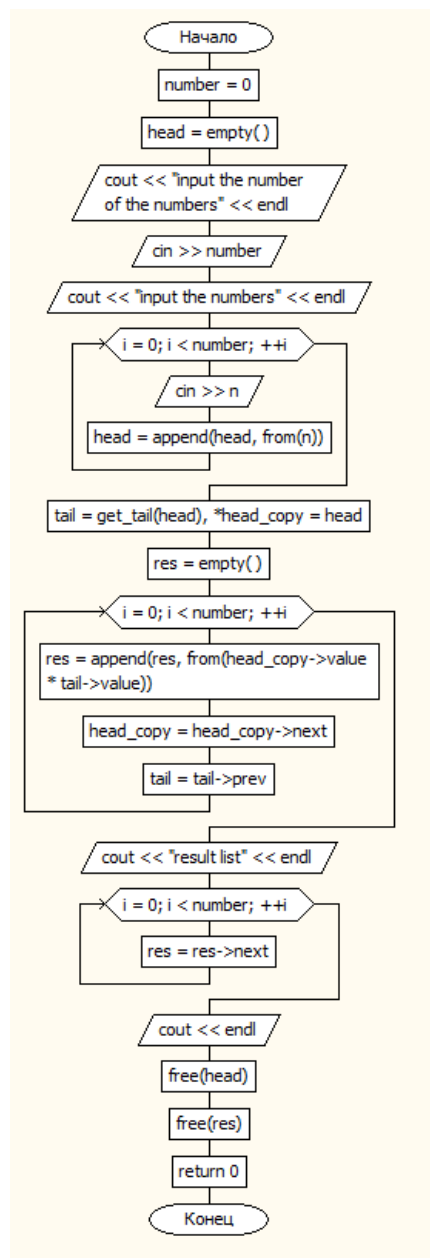


Рисунок 1 - схема алгоритма main

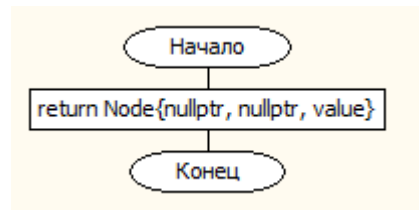


Рисунок 2 - схема алгоритма функции empty

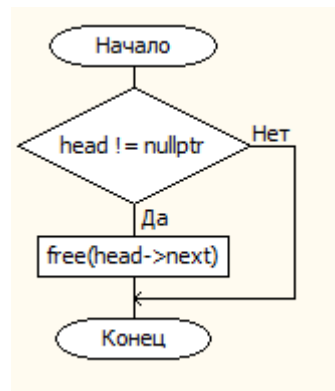


Рисунок 3 - схема алгоритма функции empty

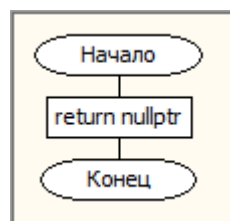


Рисунок 4 - схема алгоритма функции free

Текст программы

```
You, 13 minutes ago | 1 author (You)
1  #include <cstring>
2  #include <iostream>
3
4  using namespace std;
5
You, 13 minutes ago | 1 author (You)
6  typedef struct Node {
7      Node *next;
8      Node *prev;
9      double value;
10 } Node;
11
12 Node *empty() { return nullptr; }
13
14 Node *from(double value) { return new Node{nullptr, nullptr, value}; }
15
16 Node *append(Node *head, Node *new_node) {
17     if (head == nullptr) {
18         return new_node;
19     } else {
20         head->next = append(head->next, new_node);
21         head->next->prev = head;
22         return head;
23     }
24 }
25
26 Node *get_tail(Node *head) {
27     if (head->next != nullptr) {
28         return get_tail(head->next);
29     } else {
30         return head;
31     }
32 }
33
34 void free(Node *head) {
35     if (head != nullptr) {
36         free(head->next);
37         delete head;
38     }
39 }
```

Рисунок 5 - код программы

```

41 int main() {
42     int number = 0;
43     Node *head = empty();
44
45     cout << "input the number of the numbers" << endl;
46     cin >> number;
47     cout << "input the numbers" << endl;
48     for (int i = 0; i < number; ++i) {
49         double n;
50         cin >> n;
51         head = append(head, from(n));
52     }
53
54     Node *tail = get_tail(head), *head_copy = head;
55     Node *res = empty();
56     for (int i = 0; i < number; ++i) {
57         res = append(res, from(head_copy->value * tail->value));
58         head_copy = head_copy->next;
59         tail = tail->prev;
60     }
61
62     cout << "data list" << endl;
63     for (int i = 0; i < number; ++i) {
64         cout << head->value << " ";
65         head = head->next;
66     }
67     cout << endl;
68
69     cout << "result list" << endl;
70     for (int i = 0; i < number; ++i) {
71         cout << res->value << " ";
72         res = res->next;
73     }
74     cout << endl;
75
76     free(head);
77     free(res);
78     return 0;
79 }

```

Рисунок 6 - код программы

Тестовые данные

```

$ ./main
input the number of the numbers
5
input the numbers
1 2 3 4 5
data list
1 2 3 4 5
result list
5 8 9 8 5

```

Рисунок 7 - пример работы программы

Вывод

В лабораторной работе были изучены средства работы с динамическими структурами данных.