

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

К.А. Кочин

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

ЛИНЕЙНЫЕ И ЦИКЛИЧЕСКИЕ СПИСКИ

по курсу: Структуры и алгоритмы обработки данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

подпись, дата

Бобрович Н. С.

инициалы, фамилия

Санкт-Петербург 2022

Цель работы

Целью работы является изучение структур данных «линейный список» и «циклический список», а также получение практических навыков их реализации.

Задание

Вариант №22.

Реализовать структуры данных «линейный список» и «циклический список» в соответствии с заданным вариантом. Дополнительно программа должна осуществлять следующие операции:

- 1) Добавление/удаление элемента в список (с клавиатуры);
- 2) Вывод исходного и результирующего списков на экран;
- 3) Если списки являются многочленами, в выводе должна быть отражена степень каждого элемента.

Дана последовательность неповторяющихся чисел. Получить 2 последовательности, первая из которых является зеркальным отображением исходной. Вторая – поэлементная разница между исходной и второй последовательностью. (Список циклический двусвязный)

Листинг программы

```

1  #include <iostream>
2
3  using namespace std;
4
5  struct node {
6      int val;
7      node* next;
8  };
9
10 class List {
11     node* Head;
12 public:
13     List() {
14         Head = NULL;
15     }
16
17     ~List() {
18         while (Head != NULL) {
19             node* temp = Head->next;
20             delete Head;
21             Head = temp;
22         }
23     }
24
25     void insert(int x, int pos) {
26         if (pos == 1) {
27             node* temp = new node;
28             temp->val = x;
29             temp->next = Head;
30             Head = temp;
31         }
32         else {
33             node* temp = new node;
34             temp->val = x;
35             if (Head == NULL) {
36                 Head = temp;
37             }
38             else {
39                 node* tempNew = Head;
40                 int count = 1;
41                 while (tempNew->next != NULL && count < pos - 1) {
42                     tempNew = tempNew->next;
43                     count++;
44                 }
45                 node* tempNew2 = tempNew->next;
46                 tempNew->next = temp;
47                 temp->next = tempNew2;
48             }
49         }
50     }
51 }

```

```

50     }
51
52     void del(int pos) {
53         node* temp = Head;
54         if (pos == 1) {
55             Head = Head->next;
56             delete temp;
57         }
58         else {
59             node* del;
60             int count = 1;
61             while (Head != NULL && count < pos - 1) {
62                 temp = temp->next;
63                 count++;
64             }
65             del = temp->next;
66             temp->next = del->next;
67             delete del;
68         }
69     }

```

```

70     void print() {
71         node* temp = Head;
72         while (temp != NULL) {
73             cout << temp->val << " ";
74             temp = temp->next;
75         }
76     }
77     int call(int pos) {
78         int e, count;
79         node* temp = new node;
80         temp = Head;
81         count = 1;
82         while (count < pos) {
83             temp = temp->next;
84             count++;
85         }
86         e = temp->val;
87         return e;
88     }
89 };
90

```

```

91 int main()
92 {
93     setlocale(LC_ALL, "Rus");
94
95     List mylist, mirlist, diflist;
96     int n, count, m, a, b, k, nlast;
97     count = 1;
98
99     cout << "Введите количество элементов: ";
100    cin >> n;
101    nlast = n;
102    cout << "Вводите элементы с первого: ";
103    for (int i = 0; i < n; i++) {
104        cin >> k;
105        mylist.insert(k, count);
106        count++;
107    }
108    count = 1;
109    for (int i = 0; i < n; i++) {
110        mirlist.insert(mylist.call(n + 1 - count), count);
111        count++;
112    }
113    cout << endl;
114    cout << "Если хотите добавить элемент, нажмите 1.\nЕсли хотите удалить эл
115    cin >> m;

```

```

116    if ((m != 1) && (m != 2) && (m != 0)) {
117        cout << "Error" << endl;
118    }
119    if (m == 1) {
120        cout << "Введите значение элемента: ";
121        cin >> a;
122        cout << "Введите позицию элемента: ";
123        cin >> b;
124        mylist.insert(a, b);
125        mirlist.insert(a, n + 2 - b);
126        nlast++;
127    }
128    if (m == 2) {
129        cout << "Введите номер элемента: ";
130        cin >> b;
131        mylist.del(b);
132        mirlist.del(n + 1 - b);
133        nlast--;
134    }
135    if (m == 0) {
136        cout << "Thanks" << endl;
137    }
138

```

```
139     count = 1;
140     for (int i = 0; i < nlast; i++) {
141         diflist.insert((mylist.call(count) - mirlist.call(count)), count);
142         count++;
143     }
144
145     cout << "Исходный список ";
146     mylist.print();
147     cout << endl;
148     cout << "Список обратный исходному ";
149     mirlist.print();
150     cout << endl;
151     cout << "Результирующий список разности элементов ";
152     diflist.print();
153
154     return 0;
155 }
156
```

Контрольные примеры

```
Выбрать Консоль отладки Microsoft Visual Studio
Введите количество элементов: 5
Вводите элементы с первого: 1
2
3
4
5
Если хотите добавить элемент, нажмите 1.
Если хотите удалить элемент, нажмите 2.
Если ничего не хотите, нажмите 0.
1
Введите значение элемента: 0
Введите позицию элемента: 4
Исходный список 1 2 3 0 4 5
Список обратный исходному 5 4 0 3 2 1
Результирующий список разности элементов -4 -2 3 -3 2 4
C:\Users\User\source\repos\SAODLR2 100\x64\Debug\SAODLR2 100.exe (процесс 1196) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:

Консоль отладки Microsoft Visual Studio
Введите количество элементов: 6
Вводите элементы с первого: 1
2
3
4
5
6
Если хотите добавить элемент, нажмите 1.
Если хотите удалить элемент, нажмите 2.
Если ничего не хотите, нажмите 0.
2
Введите номер элемента: 4
Исходный список 1 2 3 5 6
Список обратный исходному 6 5 3 2 1
Результирующий список разности элементов -5 -3 0 3 5
C:\Users\User\source\repos\SAODLR2 100\x64\Debug\SAODLR2 100.exe (процесс 1196) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Вывод

Написал алгоритм, отвечающий всем требованиям задачи. Как итог, изучил структуры данных «линейный список» и «циклический список», а также получил практические навыки их реализации.