

```

1      /*
      File:      dv_spis3.cpp

      Compiler:  VC++ 3.1

      Задача.   Сформировать двусвязный список из N элементов.

      Примечание.
      Решить задачу, без использования переменной типа:
      element * * (указатель на указатель).

      Решение.  MSP,
      CNIT "North Star",
      02.11.07, 17:51
      */

12     # include <stdio.h>
13     # include <conio.h>

14     // структура одного элемента двусвязного списка
15     class element
16     {
17     public:
18         element * L; // указатель на предыдущий элемент списка
19         int data; // ключ элемента списка
20         element * R; // указатель на следующий элемент списка
21     };

22     int main ()
23     {
24         clrscr ();

25         int N; // длина двусвязного списка

26         printf ("Сколько элементов в списке чисел: ");
27         scanf ("%i", & N);
28         fflush (stdin);

29         // Под каждый элемент списка нужно будет выделить 10 байт.
30         // Нам придется N раз выделять память и тут же прописывать
31         // значения левого и правого указателей.
32         // Процесс непростой, нужно все рисовать на бумаге.

33         int i; // номер элемента двусвязного списка

34         element
35         * first, // указатель на первый элемент списка
36         * last, // указатель на последний элемент списка
37         * pred, // указатель на предыдущий элемент

```

```

38     * tek; // текущий указатель на элемент списка

39     // выделим память для самого левого элемента списка,
40     // для первого элемента списка
41     pred = new element;

42     // пропишем поля элемента, слева-направо
43     pred -> L = NULL;
44     pred -> data = 1;
45     pred -> R = NULL; // потому что еще не СОЗДАН следующий элемента

46     // запомним адрес pred как адрес первого элемента списка
47     first = pred;

48     // заполним все остальные элементы списка
49     for (i = 2; i <= N; i ++)
50     {
51         tek = new element; // найден адрес элемента списка,
52                             // значения его полей "мусорные"

53         // свяжем предыдущий элемент с текущим
54         pred -> R = tek;

55         // заполним поля элемента, слева-направо
56         tek -> L = pred;
57         tek -> data = i;
58         tek -> R = NULL;

59         // для продолжения цикла назначим
60         // текущий элемент списка предыдущим
61         pred = tek;
62     }

63     // запомним адрес pred как последний элемент списка
64     last = pred;

65     // напечатаем список в ПРЯМОМ порядке
66     printf ("\n\nСписок в прямом порядке:\n");
67     tek = first;
68     while (tek != NULL)
69     {
70         printf ("%i ", tek -> data);
71         tek = tek -> R;
72     }

73     // напечатаем список в ОБРАТНОМ порядке
74     printf ("\n\nСписок в обратном порядке:\n");
75     tek = last;
76     while (tek != NULL)
77     {
78         printf ("%i ", tek -> data);

```

```

79     tek = tek -> L;
80 }

81 // Двусвязный список занимает определенное место в памяти,
82 // вернем память OS
83 tek = last;
84 while (tek != NULL)
85 {
86     pred = tek -> L;
87     delete tek;
88     tek = pred;
89 }

90 printf (
91     "\n\nСписок уничтожен, память возвращена"
92     " в распоряжение OS..."
93 );

94 // Научились создавать и уничтожать двусвязные списки.

95 // Теперь, а как переписать этот код так, чтобы не нужно
96 // было использовать переменную pred?
97 // Как оптимизировать этот код?

98 getch ();

99     return 0;
100 }

```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-3-task-14.html>

Sergey Mitrofanov, 30.10.13, 20:57

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2013