

```
1 #ifdef _WIN32
2     #include <tchar.h>
3     #include <conio.h>
4 #elif (defined __linux__) || (defined _AIX)
5     typedef char _TCHAR;
6     #define _tmain main
7 #endif
8
9 #include <iostream>
10 using namespace std;
11
12 #include "LinkedList1.h"
13 //-----
14 TListNode::TListNode()
15 {
16     Data = 0;
17     nextPtr = NULL; // dolezite !!!!!!!!!!!!!!!!!!!!!!!
18 }
19 //-----
20 TListNode::~TListNode()
21 {
22 }
23 //-----
24 TList::TList()
25 {
26     firstPtr = lastPtr = NULL; // dolezite !!!!!!!!!!!!!!!!!!!!!!!
27 }
28 //-----
29 TList::~TList()
30 {
31     TListNode *currentPtr = firstPtr, *tempPtr;
32
33     if (!isEmpty()) {
34         cout << "\nDealokujeme uzly ...\n";
35
36         while (currentPtr != NULL) {
37             tempPtr = currentPtr;
38             cout << tempPtr->Data << endl;
39             currentPtr = currentPtr->nextPtr;
40             delete tempPtr;
41         }
42         cout << "Vsetky uzly tohto zoznamu su dealokovane.\n";
43     }
44 }
45 //-----
46 void TList::insert(int data) // vlozi prvok usporiadane od najmensieho po najvacsi
47 {
48     TListNode *newPtr = new TListNode;
49     newPtr->Data = data;
50     newPtr->nextPtr = NULL;
51
52     if (this->firstPtr == NULL)
53         this->firstPtr = this->lastPtr = newPtr;
```

```
54     else
55         if (data < this->firstPtr->Data) {
56             newPtr->nextPtr = this->firstPtr;
57             this->firstPtr = newPtr;
58         }
59         else if (data > this->lastPtr->Data) {
60             this->lastPtr->nextPtr = newPtr;
61             this->lastPtr = newPtr;
62         }
63         else {
64             TListNode *p1 = this->firstPtr, *p2 = this->firstPtr->nextPtr;
65             while (p2->Data < data) {
66                 p1 = p2;
67                 p2 = p2->nextPtr;
68             }
69             p1->nextPtr = newPtr;
70             newPtr->nextPtr = p2;
71         }
72     }
73     //-----
74     void TList::include(TList* z)
75     {
76         TListNode *p = z->firstPtr;
77
78         while (p != NULL) {
79             this->insert(p->Data);
80             p = p->nextPtr;
81         }
82     }
83     //-----
84     ostream& operator<<(ostream& os, const TList* z)
85     {
86         if (z->isEmpty()) {
87             os << "Zoznam je prazdny!" << endl;
88             return os; // ak je zoznam prazdny, koncime
89         }
90         TListNode *p = z->getFirstPtr(); // pomocny prvok typu TPrvok
91         while (p->getNextPtr() != NULL) // prechadzaj cez vsetky prvky
92         {
93             os << p->getData() << " "; // vypis datovu cast prvku
94             p = p->getNextPtr(); // chod na dalsi prvok
95         }
96         os << p->getData(); // vypis posledny prvok
97         return os;
98     }
99     //-----
100
```