5 linked lists are stored in the memory illustrated on the back of this page. Each memory address is numbered and the contents of the LNode object stored at the address are described.

If the contents of the Linked List (LList2) objects are:

- **List1**: first: 81; last: 84; size: 12; direction: FORWARD
- **List2**: first: 76; last: 9; size: 15; direction: BACKWARD
- **List3**: first: 15; last: 11; size: 18; direction: FORWARD
- **List4**: first: 67; last: 53; size: 12; direction: BACKWARD
- **List5**: first: 3; last: 25; size: 23; direction: FORWARD

What is the output of the following code segment?

```cpp
cout << "List1 : " << List1 << endl;
cout << "List2 : " << List2 << endl;
cout << "List3 : " << List3 << endl;
cout << "List4 : " << List4 << endl;
cout << "List5 : " << List5 << endl;
```
1: Memory address not used
2: prev: 46; next: 51; data: machines
3: prev: 0; next: 54; data: When
4: prev: 19; next: 22; data: you
5: prev: 71; next: 16; data: programmers
6: Memory address not used
7: prev: 85; next: 86; data: If
8: prev: 30; next: 78; data: it
9: prev: 38; next: 0; data: Programs
10: prev: 61; next: 49; data: longer
11: prev: 59; next: 60; data: that
12: prev: 0; next: 21; data: Real
13: prev: 79; next: 89; data: in
14: prev: 59; next: 60; data: a
15: prev: 41; next: 38; data: be
16: prev: 50; next: 4; data: start
17: prev: 83; next: 40; data: reluctance,
18: prev: 41; next: 38; data: be
19: prev: 50; next: 4; data: start
20: prev: 81; next: 36; data: is
21: prev: 15; next: 52; data: programmers
22: prev: 4; next: 53; data: sooner
23: prev: 82; next: 84; data: about
24: prev: 55; next: 57; data: language
25: prev: 90; next: 0; data: English.
26: prev: 66; next: 34; data: and
27: Memory address not used
28: Memory address not used
29: Memory address not used
30: prev: 43; next: 8; data: write,
31: Memory address not used
32: prev: 65; next: 68; data: will
33: prev: 51; next: 66; data: incidentally
34: prev: 26; next: 63; data: read,
35: prev: 36; next: 12; data: with
36: prev: 20; next: 35; data: anticipated
37: prev: 47; next: 41; data: for
38: prev: 18; next: 9; data: must
39: prev: 52; next: 56; data: comment
40: prev: 17; next: 82; data: and
41: prev: 37; next: 18; data: written
42: prev: 86; next: 75; data: was
43: prev: 75; next: 30; data: to
44: prev: 48; next: 45; data: hard
45: prev: 44; next: 11; data: to
46: prev: 76; next: 2; data: to
47: prev: 63; next: 37; data: people
48: prev: 78; next: 44; data: be
49: prev: 10; next: 58; data: the
50: prev: 58; next: 19; data: to
51: prev: 2; next: 33; data: for
52: prev: 21; next: 39; data: don't
53: prev: 22; next: 0; data: The
54: prev: 49; next: 35; data: is
55: prev: 54; next: 24; data: programming
56: prev: 39; next: 85; data: their
57: prev: 24; next: 59; data: is
58: prev: 49; next: 50; data: code,
59: prev: 57; next: 14; data: created
60: prev: 14; next: 62; data: allows
61: prev: 87; next: 10; data: the
62: prev: 60; next: 64; data: programmers
63: prev: 34; next: 47; data: to
64: prev: 62; next: 79; data: to
65: prev: 88; next: 32; data: it
66: prev: 33; next: 26; data: only
67: prev: 0; next: 77; data: take.
68: prev: 32; next: 70; data: be
69: prev: 12; next: 83; data: performed
70: prev: 68; next: 71; data: discovered
71: prev: 70; next: 5; data: that
72: Memory address not used
73: Memory address not used
74: Memory address not used
75: prev: 42; next: 43; data: hard
76: prev: 0; next: 46; data: execute.
77: prev: 67; next: 87; data: will
78: prev: 8; next: 48; data: should
79: prev: 64; next: 13; data: program
80: Memory address not used
81: prev: 0; next: 20; data: Debugging
82: prev: 40; next: 23; data: bragged
83: prev: 69; next: 17; data: with
84: prev: 23; next: 0; data: forever.
85: prev: 56; next: 7; data: code.
86: prev: 7; next: 42; data: it
87: prev: 77; next: 61; data: program
88: prev: 89; next: 65; data: English,
89: prev: 13; next: 88; data: simple
90: prev: 16; next: 25; data: speak