Doubly Linked List Tracing Exercise
(Homework 3 - Due: Monday, 25 April 2016)

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 LList2 <string> List1, List2, List3, List4, and List5 objects are:

List 1 : first:   27; last:   18; size:   12; direction: FORWARD
List 2 : first:    1; last:   15; size:   15; direction: BACKWARD
List 3 : first:   56; last:    2; size:   10; direction: FORWARD
List 4 : first:    5; last:   53; size:   23; direction: BACKWARD
List 5 : first:   70; last:   46; size:   17; direction: FORWARD

What is the output of the following C++ code segment?

```cpp
cout << "List1: " << List1 << endl;
cout << "List2: " << List2 << endl;
cout << "List3: " << List3 << endl;
cout << "List4: " << List4 << endl;
cout << "List5: " << List5 << endl;
```
execute.
actors.
reluctance,
written.
English.
people
cannot.
to.
is.
machines.
read,
anticipated.
incidentally.
for.
Programs.
written.
programming.
not.
about.
it.
now.
Debugging.
it's.
a.
now.
that.
created.
inventive.
playwrights.
for.
all.

Memory address not used

Memory address not used

Memory address not used


read.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.

expected.
Doubly Linked List Tracing Exercise
(Homework 3 - Due: Monday, 25 April 2016)

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 LList2 <string> List1, List2, List3, List4, and List5 objects are:

List 1 : first: 30; last: 62; size: 15; direction: FORWARD
List 2 : first: 44; last: 25; size: 23; direction: BACKWARD
List 3 : first: 73; last: 17; size: 12; direction: FORWARD
List 4 : first: 26; last: 10; size: 10; direction: BACKWARD
List 5 : first: 64; last: 45; size: 18; direction: FORWARD

What is the output of the following C++ code segment?

```cpp
cout << "List1: " << List1 << endl;
cout << "List2: " << List2 << endl;
cout << "List3: " << List3 << endl;
cout << "List4: " << List4 << endl;
cout << "List5: " << List5 << endl;
```
English program only that bragged that programmers with be to

be All be don't it was about forever.

read, written for code. to write, be when.

All: or, written for code. to write, be when.

Programming machines language for is to reluctance, in distaste, execute.

Playwrights allows programmers are their will that must.

Debugging people speak with and cannot to
Doubly Linked List Tracing Exercise
(Homework 3 - Due: Monday, 25 April 2016)

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 LList2 <string> List1, List2, List3, List4, and List5 objects are:

List 1 : first:  66; last:  26; size:  12; direction: FORWARD
List 2 : first:  18; last:  15; size:  15; direction: BACKWARD
List 3 : first:  56; last:  44; size:  12; direction: FORWARD
List 4 : first:  39; last:  79; size:  23; direction: BACKWARD
List 5 : first:  37; last:  78; size:  10; direction: FORWARD

What is the output of the following C++ code segment?

```cpp
cout << “List1: “ << List1 << endl;
cout << “List2: “ << List2 << endl;
cout << “List3: “ << List3 << endl;
cout << “List4: “ << List4 << endl;
cout << “List5: “ << List5 << endl;
```
Memory address not used

Memory address not used

prev: 14; next: 10; data: anticipated

prev: 5; next: 17; data: with

prev: 7; next: 71; data: machines

prev: 18; next: 11; data: to

prev: 10; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 37; next: 73; data: programmers

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 18; next: 11; data: to

prev: 5; next: 17; data: with

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,

prev: 57; next: 63; data: written

prev: 7; next: 71; data: machines

prev: 66; next: 5; data: is

prev: 72; next: 0; data: Programs

prev: 58; next: 58; data: distaste,

prev: 0; next: 7; data: execute.

prev: 50; next: 29; data: read,