

```

1 // Filename: pointers.cpp
2 // Source: Phil-ads (unpublished class notes in Fall 2014 by Phil Pratt-Szeliga)
3 #include <iostream>
4
5 struct SingleNode {
6     SingleNode * m_next;
7     long long m_value;
8 };
9
10 void printAddrValues(SingleNode * node, long long addr, const char * title){
11     std::cout << "=====" << std::endl;
12     std::cout << "= " << title << " addrs/values: " << std::endl;
13     std::cout << "=====" << std::endl;
14     std::cout << "addr(" << title << "): " << std::hex << addr << std::endl;
15     std::cout << "value(" << title << "): " << std::hex << node << std::endl;
16     std::cout << "addr(" << title << ".m_next): " << std::hex << &(node->m_next) << std::endl;
17     std::cout << "value(" << title << ".m_next): " << std::hex << node->m_next << std::endl;
18     std::cout << "addr(" << title << ".m_value): " << std::hex << &(node->m_value) << std::endl;
19     std::cout << "value(" << title << ".m_value): " << std::hex << node->m_value << std::endl;
20     std::cout << std::endl;
21 }
22
23 int main(int argc, char *argv[]){
24
25     std::cout << "=====" << std::endl;
26     std::cout << "= sizes: " << std::endl;
27     std::cout << "=====" << std::endl;
28     std::cout << "sizeof(SingleNode): " << sizeof(SingleNode) << std::endl;
29     std::cout << "sizeof(SingleNode *): " << sizeof(SingleNode *) << std::endl;
30     std::cout << "sizeof(long long): " << sizeof(long long) << std::endl;
31     std::cout << std::endl;
32
33     SingleNode * head = new SingleNode();
34     printAddrValues(head, (long long) &head, "head");
35
36     SingleNode * one = new SingleNode();
37     one->m_value = 1;
38
39     SingleNode * two = new SingleNode();
40     two->m_value = 2;
41
42     SingleNode * three = new SingleNode();
43     three->m_value = 3;
44
45     head->m_next = one;
46     one->m_next = two;
47     two->m_next = three;
48     printAddrValues(head, (long long) &head, "head");
49     printAddrValues(one, (long long) &one, "one");
50     printAddrValues(two, (long long) &two, "two");
51     printAddrValues(three, (long long) &three, "three");
52
53     return 0;
54 }

```