```cpp
// Filename: pointers.cpp
// Source: Phil-ads (unpublished class notes in Fall 2014 by Phil Pratt-Szeliga)
#include <iostream>

struct SingleNode {
  SingleNode * m_next;
  long long m_value;
};

void printAddrValues(SingleNode * node, long long addr, const char * title){
  std::cout << "================================================" << std::endl;
  std::cout << "= " << title << " addrs/values: " << std::endl;
  std::cout << "================================================" << std::endl;
  std::cout << "addr(" << title << "): " << std::hex << addr << std::endl;
  std::cout << "value(" << title << "): " << std::hex << node << std::endl;
  std::cout << "addr(" << title << ".m_next): " << std::hex << &(node->m_next) << std::endl;
  std::cout << "value(" << title << ".m_next): " << std::hex << node->m_next << std::endl;
  std::cout << "addr(" << title << ".m_value): " << std::hex << &(node->m_value) << std::endl;
  std::cout << "value(" << title << ".m_value): " << std::hex << node->m_value << std::endl;
  std::cout << std::endl;
}

int main(int argc, char *argv[]){

  std::cout << "================================================" << std::endl;
  std::cout << "= sizes: " << std::endl;
  std::cout << "================================================" << std::endl;
  std::cout << "sizeof(SingleNode): " << sizeof(SingleNode) << std::endl;
  std::cout << "sizeof(SingleNode *): " << sizeof(SingleNode *) << std::endl;
  std::cout << "sizeof(long long): " << sizeof(long long) << std::endl;
  std::cout << std::endl;

  SingleNode * head = new SingleNode();
  printAddrValues(head, (long long) &head, "head");

  SingleNode * one = new SingleNode();
  one->m_value = 1;

  SingleNode * two = new SingleNode();
  two->m_value = 2;

  SingleNode * three = new SingleNode();
  three->m_value = 3;

  head->m_next = one;
  one->m_next = two;
  two->m_next = three;
  printAddrValues(head, (long long) &head, "head");
  printAddrValues(one, (long long) &one, "one");
  printAddrValues(two, (long long) &two, "two");
  printAddrValues(three, (long long) &three, "three");

  return 0;
}
```