

```
1: // $Id: listfree.cpp,v 1.20 2015-03-26 22:10:36-07 - - $
2:
3: // Show how to break a cycle in a simple circular list.
4:
5: #include <algorithm>
6: #include <iostream>
7: #include <memory>
8: using namespace std;
9:
10: struct node;
11:
12: using node_ptr = shared_ptr<node>;
13:
14: struct node {
15:     int value;
16:     node_ptr link;
17:     node (int value, node_ptr link): value(value), link(link) {}
18: };
19:
20: int main (int argc, char** argv) {
21:     cout << "Command:";
22:     for_each (&argv[0], &argv[argc], [](char* arg){cout << " " << arg;});
23:     cout << endl;
24:     bool break_cycle = argc > 1 and string (argv[1]) == "-f";
25:     node_ptr list = make_shared<node> (1,
26:                                     make_shared<node> (2,
27:                                     make_shared<node> (3, nullptr)));
28:     list->link->link->link = list;
29:     cout << "list = " << list << endl;
30:     for (auto curr = list;;) {
31:         cout << curr << " -> {" << curr->value << ", " << curr->link
32:             << "} (use_count " << curr.use_count() << ")" << endl;
33:         curr = curr->link;
34:         if (curr == list) break;
35:     }
36:     if (break_cycle) list->link = nullptr;
37:     return 0;
38: }
39:
40: //TEST// valgrind listfree -0 >listfree.out-0 2>&1
41: //TEST// valgrind listfree -f >listfree.out-f 2>&1
42: //TEST// mkpspdf listfree.ps listfree.cpp* listfree.out-*
43:
```

[illegible]

```
1: ==23319== Memcheck, a memory error detector
2: ==23319== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al
.
3: ==23319== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright
info
4: ==23319== Command: listfree -0
5: ==23319==
6: Command: listfree -0
7: list = 0x9c9a1e0
8: 0x9c9a1e0 -> {1, 0x9c9a170} (use_count 3)
9: 0x9c9a170 -> {2, 0x9c9a100} (use_count 2)
10: 0x9c9a100 -> {3, 0x9c9a1e0} (use_count 2)
11: ==23319==
12: ==23319== HEAP SUMMARY:
13: ==23319==      in use at exit: 120 bytes in 3 blocks
14: ==23319==    total heap usage: 5 allocs, 2 frees, 163 bytes allocated
15: ==23319==
16: ==23319== LEAK SUMMARY:
17: ==23319==      definitely lost: 40 bytes in 1 blocks
18: ==23319==      indirectly lost: 80 bytes in 2 blocks
19: ==23319==      possibly lost: 0 bytes in 0 blocks
20: ==23319==      still reachable: 0 bytes in 0 blocks
21: ==23319==           suppressed: 0 bytes in 0 blocks
22: ==23319== Rerun with --leak-check=full to see details of leaked memory
23: ==23319==
24: ==23319== For counts of detected and suppressed errors, rerun with: -v
25: ==23319== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 1 from 1)
```

```
1: ==23354== Memcheck, a memory error detector
2: ==23354== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al
.
3: ==23354== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright
info
4: ==23354== Command: listfree -f
5: ==23354==
6: Command: listfree -f
7: list = 0x9c9a1e0
8: 0x9c9a1e0 -> {1, 0x9c9a170} (use_count 3)
9: 0x9c9a170 -> {2, 0x9c9a100} (use_count 2)
10: 0x9c9a100 -> {3, 0x9c9a1e0} (use_count 2)
11: ==23354==
12: ==23354== HEAP SUMMARY:
13: ==23354==      in use at exit: 0 bytes in 0 blocks
14: ==23354==    total heap usage: 5 allocs, 5 frees, 163 bytes allocated
15: ==23354==
16: ==23354== All heap blocks were freed -- no leaks are possible
17: ==23354==
18: ==23354== For counts of detected and suppressed errors, rerun with: -v
19: ==23354== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 1 from 1)
```