The accompanying C file implements a doubly-linked list with integer payloads. The central function of the program (gl_sort) sorts the list using a bubble sort-style algorithm in ascending order of payloads. The datat structure also supports nesting, but this is not used for sorting. Separation between nesting pointers and list linkage pointers should be maintained though.

Verify the program w.r.t. this informal specification. Should you find bugs, please fix them and proceed to verify. If your tool does not support C, we ask you to reimplement the core data structure/functionality in the language of your choice. Please try to stay as faitful to the original code as possible (but see below).

The program contains a basic correctness checker consisting of assert statements (inspect function). The checker does not check the complete specification given above.

* For tools that do not need user-supplied invariants, checking that the assertions pass would be a starting point. Feel free to add assertions that are meaningful w.r.t. the informal specification. If you can verify one of the included assertions, please comment it out to produce a maximal set of assertions that you can verify.

* For tools that use more expressive specification formalisms and user-supplied invariants, we encourage you to prove a more complete functional specification rather than just checking the assertions.

A partial and approximate overview of the data structure is below (please consider the text below as well).

```
+——————————————————————————————————————————————————————————————————+
|                                        |
|                                        |
|                                        |
|             +———————————+     +———————————+      |
|    gl_list     |  value   |     |  value   |     |
|  +—————————+    +———————————+     +———————————+      |
+————> next  +——————>  next   +———————>  next   +—————————+
   +—————————+    +———————————+     +———————————+ linkage
   |  prev   |    |  prev    |     |  prev    |
   +—————————+    +———————————+     +———————————+
             |  next    |     |  next    |
             |-------------|     |-------------| nested
             |  prev    |     |  prev    |
             |       |     |       |
             |       |     |       |
             |       |     |       |
             +———————————+     +———————————+
```

Implementation details:

The next and the prev pointers are contained in a substructure called list_head. They point to the list_head of the neighboring nodes. From this structure, the payload (i.e., value) is reached by means of pointer arithmetic (this is an aspect that need not be modeled in other languages).

Please send solutions per email to:
the seminar organizers