```cpp
 1: // $Id: shortstring.cpp,v 1.25 2016-08-03 12:54:41-07 - - $
 2:
 3: // Simple short string optimization.
 4: // Works only on a little-endian machine.
 5:
 6: #include <iostream>
 7: using namespace std;
 8:
 9: struct opt_string {
10:
11:     struct long_string {
12:         size_t size_; // stores size << 1 | 1.
13:         size_t capacity_;
14:         char* data_;
15:     };
16:
17:     struct short_string {
18:         unsigned char size_ {0}; // stores size << 1.
19:         char data_[sizeof (long_string) - 1] {};
20:     };
21:
22:     union {
23:         long_string long_;
24:         short_string short_;
25:     };
26:
27:     bool is_long() const {
28:         return short_.size_ & 1;
29:     }
30:
31:     size_t size() const {
32:         return is_long() ? short_.size_ >> 1 : long_.size_;
33:     }
34:
35:     size_t capacity() const {
36:         return is_long() ? long_.capacity_ >> 1 : sizeof short_.data_ - 1;
37:     }
38:
39:     opt_string() { short_.size_ = short_.data_[0] = 0; }
40:
41:     ~opt_string() {
42:         if (is_long() and long_.data_) delete[] long_.data_;
43:     }
44:
45: };
46:
```

```
47:
48: int offset (void* base, void* field) {
49:     return static_cast<char*> (field) - static_cast<char*> (base);
50: }
51:
52: #define SHOW(X) cout << #X << " = " << X << endl;
53: int main() {
54:     opt_string u;
55:     SHOW (sizeof u);
56:     SHOW (sizeof u.long_);
57:     SHOW (sizeof u.long_.capacity_);
58:     SHOW (sizeof u.long_.size_);
59:     SHOW (sizeof u.long_.data_);
60:     SHOW (sizeof u.short_);
61:     SHOW (sizeof u.short_.size_);
62:     SHOW (sizeof u.short_.data_);
63:     SHOW (offset (&u, &u));
64:     SHOW (offset (&u, &u.long_));
65:     SHOW (offset (&u, &u.long_.capacity_));
66:     SHOW (offset (&u, &u.long_.size_));
67:     SHOW (offset (&u, &u.long_.data_));
68:     SHOW (offset (&u, &u.short_));
69:     SHOW (offset (&u, &u.short_.size_));
70:     SHOW (offset (&u, &u.short_.data_));
71: }
72:
73: //TEST// ./shortstring >shortstring.out
74: //TEST// mkpspdf shortstring.ps shortstring.cpp* shortstring.out
75:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting shortstring.cpp
    2: shortstring.cpp: $Id: shortstring.cpp,v 1.25 2016-08-03 12:54:41-07 - -
$
    3: g++ -g -O0 -std=gnu++14 -rdynamic -Wall -Wextra -Wold-style-cast shortst
ring.cpp -o shortstring -lglut -lGLU -lGL -lX11 -lrt -lm
    4: cpplint.py.perl shortstring.cpp
    5: Done processing shortstring.cpp
    6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished shortstring.cpp
```

```
 1: sizeof u = 24
 2: sizeof u.long_ = 24
 3: sizeof u.long_.capacity_ = 8
 4: sizeof u.long_.size_ = 8
 5: sizeof u.long_.data_ = 8
 6: sizeof u.short_ = 24
 7: sizeof u.short_.size_ = 1
 8: sizeof u.short_.data_ = 23
 9: offset (&u, &u) = 0
10: offset (&u, &u.long_) = 0
11: offset (&u, &u.long_.capacity_) = 8
12: offset (&u, &u.long_.size_) = 0
13: offset (&u, &u.long_.data_) = 16
14: offset (&u, &u.short_) = 0
15: offset (&u, &u.short_.size_) = 0
16: offset (&u, &u.short_.data_) = 1
```