```
 1: // $Id: numlimits.cpp,v 1.10 2013-08-20 20:03:02-07 - - $
 2:
 3: #include <iomanip>
 4: #include <iostream>
 5: #include <limits>
 6: #include <typeinfo>
 7:
 8: #include <cxxabi.h>
 9:
10: using namespace std;
11:
12: template <typename T>
13: void show (const string &label) {
14:    const char *const mangled = typeid (T).name();
15:    int status;
16:    char *unmangled = abi::__cxa_demangle (mangled, 0, 0, &status);
17:    cout << label << ":" << endl
18:         << unmangled << "(" << mangled << ") [" << sizeof (T) << "] "
19:         << setprecision (numeric_limits<T>::digits10)
20:         << numeric_limits<T>::min() << " .. "
21:         << numeric_limits<T>::max() << endl;
22:    free (unmangled); // C code allocated by malloc.
23:
24: }
25:
26: #define SHOW(T) show<T>(#T)
27:
28: int main() {
29:    SHOW (bool);
30:    SHOW (char);
31:    SHOW (signed char);
32:    SHOW (unsigned char);
33:    SHOW (short);
34:    SHOW (signed short);
35:    SHOW (unsigned short);
36:    SHOW (int);
37:    SHOW (signed int);
38:    SHOW (unsigned int);
39:    SHOW (long);
40:    SHOW (signed long);
41:    SHOW (unsigned long);
42:    SHOW (long long);
43:    SHOW (signed long long);
44:    SHOW (unsigned long long);
45:    SHOW (float);
46:    SHOW (double);
47:    SHOW (long double);
48:    return 0;
49: }
50:
51: //TEST// ./numlimits >numlimits.out 2>&1
52: //TEST// mkpspdf numlimits.ps numlimits.cpp* numlimits.out*
53:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting numlimits.cpp
    2: numlimits.cpp:
    3:       $Id: numlimits.cpp,v 1.10 2013-08-20 20:03:02-07 - - $
    4: g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++11 numlimits.cpp -o numlimi
ts -lglut -lGLU -lGL -lX11 -lrt -lm
    5: rm -f numlimits.o
    6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished numlimits.cpp
```

```
    1: bool:
    2: bool(b) [1] 0 .. 1
    3: char:
    4: char(c) [1] \200 .. \177
    5: signed char:
    6: signed char(a) [1] \200 .. \177
    7: unsigned char:
    8: unsigned char(h) [1] \000 .. ÿ
    9: short:
   10: short(s) [2] -32768 .. 32767
   11: signed short:
   12: short(s) [2] -32768 .. 32767
   13: unsigned short:
   14: unsigned short(t) [2] 0 .. 65535
   15: int:
   16: int(i) [4] -2147483648 .. 2147483647
   17: signed int:
   18: int(i) [4] -2147483648 .. 2147483647
   19: unsigned int:
   20: unsigned int(j) [4] 0 .. 4294967295
   21: long:
   22: long(l) [8] -9223372036854775808 .. 9223372036854775807
   23: signed long:
   24: long(l) [8] -9223372036854775808 .. 9223372036854775807
   25: unsigned long:
   26: unsigned long(m) [8] 0 .. 18446744073709551615
   27: long long:
   28: long long(x) [8] -9223372036854775808 .. 9223372036854775807
   29: signed long long:
   30: long long(x) [8] -9223372036854775808 .. 9223372036854775807
   31: unsigned long long:
   32: unsigned long long(y) [8] 0 .. 18446744073709551615
   33: float:
   34: float(f) [4] 1.17549e-38 .. 3.40282e+38
   35: double:
   36: double(d) [8] 2.2250738585072e-308 .. 1.79769313486232e+308
   37: long double:
   38: long double(e) [16] 3.3621031431120935l-4932 .. 1.1897314953572317l+49
32
```