```cpp
 1: // $Id: string2int.cpp,v 1.15 2014-06-20 22:46:10-07 - - $
 2:
 3: //
 4: // NAME
 5: //     str2int-stream - print out argv elements that are ints
 6: //
 7: // SYNOPSIS
 8: //     str2int-stream [string...]
 9: //
10: // DESCRIPTION
11: //     Prints out its arguments that are ints, and which are not.
12: //
13:
14: #include <iostream>
15: #include <sstream>
16: #include <stdexcept>
17: #include <string>
18: #include <vector>
19: using namespace std;
20:
21: int from_string (const string &arg) {
22:    stringstream stream (arg);
23:    int result;
24:    if (not (stream >> result and stream.eof())) {
25:       throw domain_error ("int from_string (" + arg + ")");
26:    }
27:    return result;
28: }
29:
30: int main (int argc, char **argv) {
31:    cout << boolalpha;
32:    vector<string> args (&argv[0], &argv[argc]);
33:    for (const auto& arg: args) {
34:       cout << "\"" << arg << "\"" << endl;
35:       try {
36:          int number = from_string (arg);
37:          cout << "        success: number = " << number << endl;
38:       }catch (const domain_error &error) {
39:          cout << "        failure: " << error.what() << endl;
40:       }
41:    }
42:    return 0;
43: }
44:
45: /*
46: //TEST// ./string2int 1024 56foobar -45 68.9 testing 007777 0x48 \
47: //TEST//         2147483648 2>&1 >string2int.output
48: //TEST// mkpspdf string2int.ps string2int.cpp* string2int.output
49: */
50:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting string2int.cpp
    2: string2int.cpp:
    3:      $Id: string2int.cpp,v 1.15 2014-06-20 22:46:10-07 - - $
    4: g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++11 string2int.cpp -o string
2int -lglut -lGLU -lGL -lX11 -lrt -lm
    5: rm -f string2int.o
    6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished string2int.cpp
```

```
 1: "./string2int"
 2:         failure: int from_string (./string2int)
 3: "1024"
 4:         success: number = 1024
 5: "56foobar"
 6:         failure: int from_string (56foobar)
 7: "-45"
 8:         success: number = -45
 9: "68.9"
10:         failure: int from_string (68.9)
11: "testing"
12:         failure: int from_string (testing)
13: "007777"
14:         success: number = 7777
15: "0x48"
16:         failure: int from_string (0x48)
17: "2147483648"
18:         failure: int from_string (2147483648)
```