```
 1: // $Id: findmap.cpp,v 1.4 2014-11-20 16:14:03-08 - - $
 2:
 3: #include <iostream>
 4: #include <string>
 5: #include <unordered_map>
 6: #include <vector>
 7: using namespace std;
 8:
 9: int main() {
10:    unordered_map<string,int> um;
11:    for (const string& s: vector<string> {"foo", "bar", "baz"}) {
12:       const auto& e = um.find (s);
13:       cout << s << ": ";
14:       if (e == um.end()) cout << "not found";
15:                     else cout << e->second;
16:       cout << endl;
17:    }
18:    int i{};
19:    for (const string& s: vector<string> {"foo", "bar", "baz"}) {
20:       um.insert ({s, ++i});
21:    }
22:    for (const auto& i: um) {
23:       cout << i.first << ": " << i.second << endl;
24:    }
25:    return 0;
26: }
```

```
 1: // $Id: insertmap.cpp,v 1.4 2014-11-20 16:01:18-08 - - $
 2:
 3: #include <iostream>
 4: #include <string>
 5: #include <unordered_map>
 6: #include <vector>
 7: using namespace std;
 8:
 9: int main() {
10:    unordered_map<string,int> um;
11:    for (const string& s: vector<string> {"foo", "bar", "baz"}) {
12:       cout << s << ": " << um[s] << endl;
13:    }
14:    for (const auto& i: um) {
15:       cout << i.first << ": " << i.second << endl;
16:    }
17:    return 0;
18: }
```

```
 1: // $Id: symbol-table-code.cpp,v 1.6 2015-05-13 14:40:56-07 - - $
 2:
 3: #include <bitset>
 4: #include <string>
 5: #include <unordered_map>
 6: #include <vector>
 7: using namespace std;
 8:
 9: enum { ATTR_void, ATTR_bool, ATTR_char, ATTR_int, ATTR_null,
10:        ATTR_string, ATTR_struct, ATTR_array, ATTR_function,
11:        ATTR_variable, ATTR_field, ATTR_typeid, ATTR_param,
12:        ATTR_lval, ATTR_const, ATTR_vreg, ATTR_vaddr,
13:        ATTR_bitset_size,
14: };
15: using attr_bitset = bitset<ATTR_bitset_size>;
16:
17: struct symbol;
18: using symbol_table = unordered_map<string*,symbol*>;
19: using symbol_entry = symbol_table::value_type;
20:
21: struct symbol {
22:    attr_bitset attributes;
23:    symbol_table* fields;
24:    size_t filenr, linenr, offset;
25:    size_t blocknr;
26:    vector<symbol*>* parameters;
27: };
28:
```