

CSE 6708 - Semantic Web

Techninal Documentation on Assignment 1

Paper Name: Source Code Plagiarism Detection Method Using Protégé
Built Ontologies

SAMIDHYA SARKER
Student No. 1018052049
Group-2

September 6, 2019

Contents

1	Introduction	4
1.1	Goals	4
2	Emulating Experiments of Paper Authors	5
2.1	Creating ontologies	5
2.1.1	Tools Used	5
2.1.2	Source Code	5
2.1.3	Ontologies	7
2.2	SPARQL Query	7
2.2.1	Tools Used	7
2.3	Topic Maps	10
2.3.1	Tools Used	10
2.4	Conclusion	12
3	Implementation of specified further works of the paper	13
3.1	An existing RDFized parser generator for the JAVA programming language: Codeontology	14
3.2	Creating a Parser generator using ANTLR	15
3.3	Modifying Flex/Bison Code Generators to create RDF/XML generators	17
4	Appendix	18
4.1	Owl source code for C source code defined in 2.1	18
4.2	Owl source code for JavaScript source code defined in 2.2	31

List of Figures

2.1	Topic map ontology created by source code written in 2.1 in C language	10
2.2	Topic map ontology created by source code written in 2.2 in JS	11
3.1	Topic map ontology created by source code written in 3.1 in JAVA programming language	15
3.2	ANTLR parse tree created from C source at 2.1	16
3.3	ANTLR parse tree created from JS source at 2.2	16

Listings

2.1	C source code for the max out of 3 program	6
2.2	JavaScript source code for the max out of 3 program	7
2.3	SPARQL Queries given by the paper authors	8
3.1	Java source code for the max out of 3 program	14
4.1	Owl source code for C source code defined in 2.1	18
4.2	Owl source code for JS source code defined in 2.2	31

1

Introduction

Software Plagiarism is defined as Copying Software without giving attribution. Ion Smeureanu and Bogdan Iancu of the The Bucharest University of Economic Studies have written a scientific paper. In this paper, the authors devised a use of semantic web technologies to prevent software

1.1 Goals

1. Matching authors software implementation:
 - Create ontologies from source code manually by hand using protege.
 - Execute sparql queries on ontologies and compare the metrics.
 - Create topic maps using Protege OntoGraf plugin.
2. Doing further works as dictated by the authors.
 - Create a parser.

2

Emulating Experiments of Paper Authors

2.1 Creating ontologies

2.1.1 Tools Used

- Protege 5.5.0 with OWL Code Generation Plug-in (2.0.0)
- VIM 8.1 with niklasl/vim-rdf and n3.vim Plug-in

2.1.2 Source Code

The authors of the paper had given two source codes in the paper that is identical in resulting output but different lexically. The source code takes 3 numbers as input and outputs the maximum.

One is in the C programming language.

```
1 #include <stdio.h>
2
3 int option = 0;
4 int i;
5 int numbers[3];
6
7 while (option!=3){
8     printf("Please choose an option and press enter:\n");
9     printf("1. Read 3 numbers\n 2. Print the max\n 3.Exit\n");
10    scanf("%i",&option);
11    if (option==1) {
12        for (i=0; i<3; i++) {
13            printf("\nnumbers [%i]=",i+1);
14            scanf("%i",&numbers[i]);
15        }
16    } else if (option==2) {
17        int max = 0;
18        for (i=0; i<3; i++) {
19            if(numbers[i] > max) {
20                max = numbers[i];
21            }
22        }
23        printf("\nMax=%i",max);
24    }
25 }
```

Listing 2.1: C source code for the max out of 3 program

And another is in the Javascript programming language.

```
1 var option = 0;
2 var i = 0;
3 var numbers = new Array();
4
5 while (option!=3){
6     document.write("Please choose an option and press enter:\n"
7     ↵ );
8     document.write("1. Read 3 numbers\n 2. Print the max\n 3.
9     ↵ Exit\n");
10    option = prompt("Option");
11    if (option == 1) {
12        for (i=0; i<3; i++) {
13            numbers[i] = prompt("numbers[" + (i+1) + "]");
14        }
15    } else if (option == 2) {
16        var max = 0;
17        for (i=0; i<3; i++) {
18            if(numbers[i] > max) {
19                max = numbers[i];
20            }
21        }
22        document.write("\nMax=" + max + "\n");
23    }
24 }
```

Listing 2.2: JavaScript source code for the max out of 3 program

2.1.3 Ontologies

We convert the source code into schema ontologies by defining Classes, object properties, data properties and individual code elements. The basic schema was given by authors and we extended it.

We used RDF/XML OWL format for representing source ontologies because Protege is better suited for this than RDF n3. We get ontology 4.1 for the source code defined in 2.1.

We also got a OWL ontology 4.2 created from our javascript source code defined in 2.2

2.2 SPARQL Query

2.2.1 Tools Used

- Protege 5.5.0 with SPARQL Query Plug-in (3.0.0)
- VIM 8.1 with omer/vim-sparql Plug-In

We used SPARQL Plug-In for Protege for executing sparql query in ontology

4.1 and 4.2. Instead we could have used Apache Jena Fuseki server ¹ for creating a SPARQL API endpoint.

We ran the following 8 SPARQL queries separately and got the expected results.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5
6 # For C source code
7 BASE <https://semantic-web.netlify.com/c_source_ontology/>
8
9 # For Javascript Code
10 BASE <https://semantic-web.netlify.com/js_source_ontology/>
11
12
13 # 1. Total number of conditional structures
14
15 SELECT (COUNT(?subject) AS ?c) WHERE {
16     ?subject rdf:type <ConditionalStructure>
17 }
18
19 # Expected Result on C code: 3
20 # Expected Result on JS code: 3
21
22 # 2. Total number of repetitive structures
23
24 SELECT (COUNT(?subject) AS ?c) WHERE {
25     ?subject rdf:type <RepetitiveStructure>
26 }
27
28 # Expected Result on C code: 3
29 # Expected Result on JS code: 3
30
31 # 3. Total number of variables
32
33 SELECT (COUNT(?subject) AS ?c) WHERE {
34     ?subject rdf:type <Variable>
35 }
36
37
38 # Expected Result on C code: 4
39 # Expected Result on JS code: 4
40
41 # 4. Total number of conditional structures included in
42     ↪ repetitive structures
43
44 SELECT (COUNT(?subject) AS ?c) WHERE {
45     ?object rdf:type <RepetitiveStructure> .
46     ?subject <is_included_in> ?object .
```

¹<https://jena.apache.org/>

```

47     ?subject rdf:type <ConditionalStructure>
48 }
49
50 # Expected Result on C code: 3
51 # Expected Result on JS code: 3
52
53 # 5. Total number of repetitive structures included in
54   ↪ repetitive structures
55
56 SELECT (COUNT(?subject) AS ?c) WHERE {
57     ?object rdf:type <RepetitiveStructure> .
58     ?subject <is_included_in> ?object .
59     ?subject rdf:type <RepetitiveStructure>
60 }
61
62 # Expected Result on C code: 0
63 # Expected Result on JS code: 0
64
65 # 6. Total number of system functions called
66
67 SELECT (COUNT(?subject) AS ?c) WHERE {
68     ?subject rdf:type <SystemFunction>
69 }
70
71 # Expected Result on C code: 5
72 # Expected Result on JS code: 4
73
74 # 7. Total number of system functions called in conditional
75   ↪ structures
76
77 SELECT (COUNT(?subject) AS ?c) WHERE {
78     ?object rdf:type <ConditionalStructure> .
79     ?subject <is_included_in> ?object .
80     ?subject rdf:type <SystemFunction>
81 }
82
83 # Expected Result on C code: 1
84 # Expected Result on JS code: 1
85
86 # 8. Total Number of repetative structures
87
88 SELECT (COUNT(?subject) AS ?c) WHERE {
89     ?object rdf:type <RepetitiveStructure> .
90     ?subject <is_included_in> ?object .
91     ?subject rdf:type <SystemFunction>
92 }
93
94 # Expected Result on C code: 4
95 # Expected Result on JS code: 3

```

Listing 2.3: SPARQL Queries given by the paper authors

We matched the results given by the authors concluded the experiment as a success.

2.3 Topic Maps

2.3.1 Tools Used

- Protege 5.5.0 with OntoGraf Plug-in (2.0.3)

For creating topic maps of the object properties of found ontologies, we used the OntoGraf Plug-In. We only showed the individual nodes not the class nodes and the properties associated with them as edges between nodes. We got the following map of the C source ontology at 4.1 found from source 2.1.

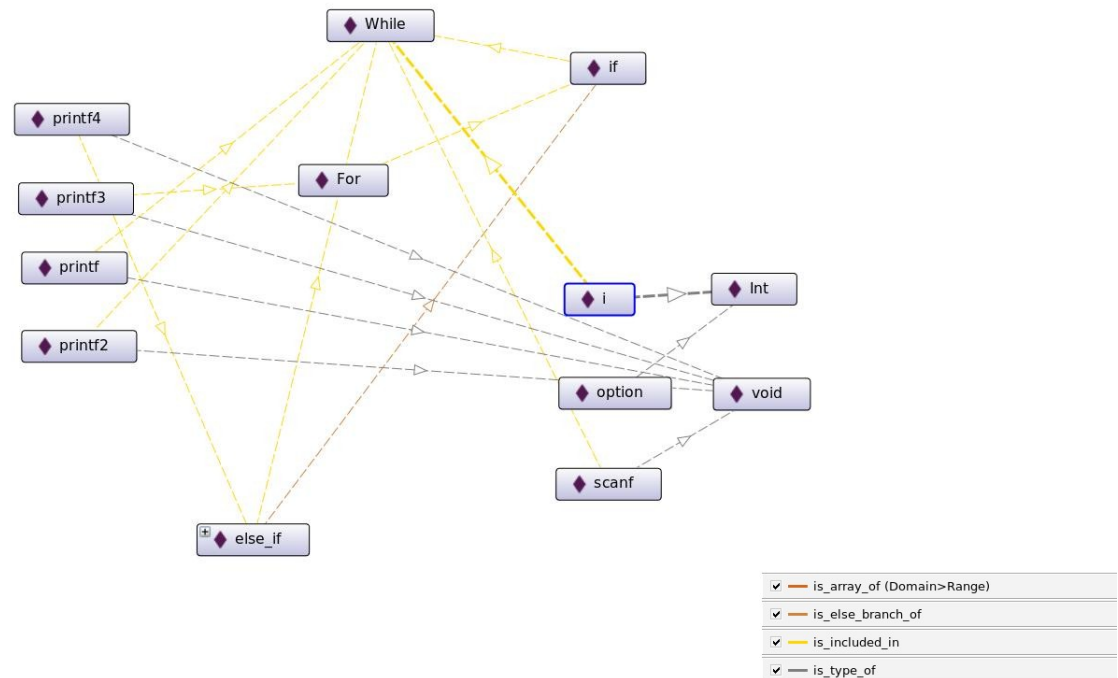


Figure 2.1: Topic map ontology created by source code written in 2.1 in C language

In the same way we got the topic maps for the source 2.2 for javascript language.

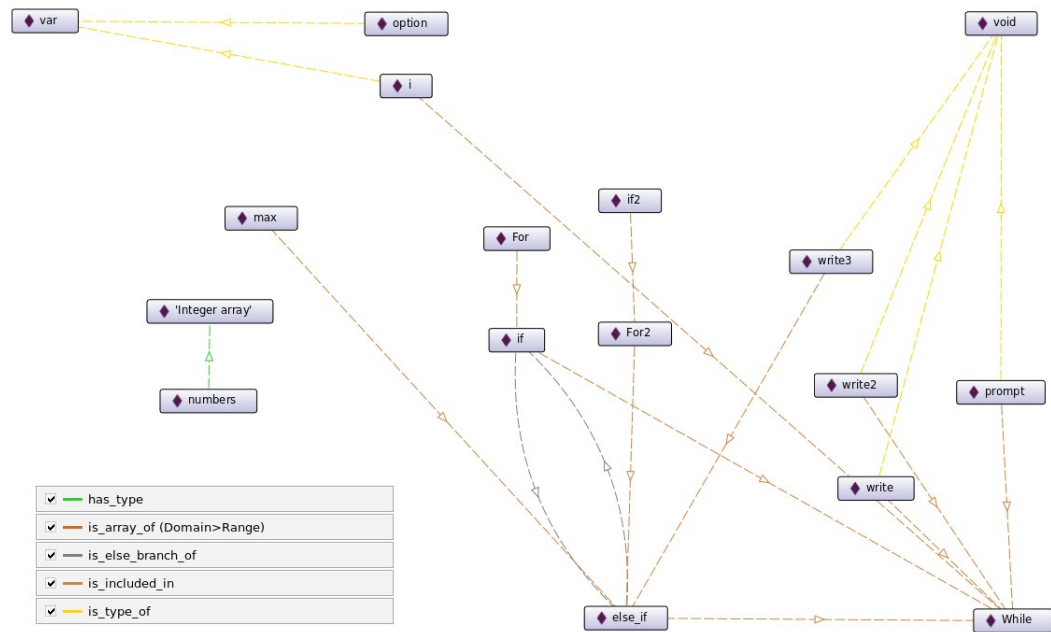


Figure 2.2: Topic map ontology created by source code written in 2.2 in JS

2.4 Conclusion

Comparing the SPARQL query results and topic maps we can decide that the source code of 2.1 and 2.2 has been plagiarized one from another.

3

Implementation of specified further works of the paper

In the further works section of the stated paper, the authors highlighted the following works that could be done:

1. A parser for code.
2. A crawler for parsing code.
3. Dynamically generated SPARQL query for defined metrics.

The creation of parser is the first step for the creation of an automated plagiarism detection system as envisioned by the authors. So, my research was focused on the formulation of a parser generator.

3.1 An existing RDFized parser generator for the JAVA programming language: Codeontology

The paper assigned to group-1¹ ² already provides a RDF triple generating parser³. So for testing if it can cater to our needs, I transliterated out source 2.1 and 2.2 into Java programming language listed 3.1

```
1 import java.util.Scanner;
2 public class MaxOfThreeNumbers {
3     public static void main(String[] args) {
4         int option = 0;
5         int i;
6         int[] numbers = new int[3];
7
8         while (option!=3){
9             System.out.println("Please choose an option and
↪ press enter:\n");
10            System.out.println("1. Read 3 numbers\n 2. Print
↪ the max\n 3.Exit\n");
11
12            Scanner scan = new Scanner(System.in);
13            option = scan.nextInt();
14
15            if (option==1) {
16                for (i=0; i<3; i++) {
17                    System.out.printf("\nnumbers [%d]=",i+1);
18                    numbers[i] = scan.nextInt();
19                }
20            } else if (option==2) {
21                int max = 0;
22                for (i=0; i<3; i++) {
23                    if(numbers[i] > max) {
24                        max = numbers[i];
25                    }
26                }
27                System.out.printf("\nMax=%d",max);
28            }
29        }
30
31    }
32 }
33 }
```

Listing 3.1: Java source code for the max out of 3 program

The main goal of the Codeontology project is to generate knowledge base for complex Java projects that is different from our goal of detecting minute information of source codes. It does not take account of programming structures, only local variables, system functions and informations related to Java

¹CodeOntology: RDF-ization of Source Code

²<http://codeontology.org/>

³<https://github.com/codeontology/parser>

programming environments like Package and Streams etc. This project can be modified to our liking and can also be adapted to other programming languages like C, Javascript.

We get a highly complex topic map different that 2.3.1 and 2.3.1.

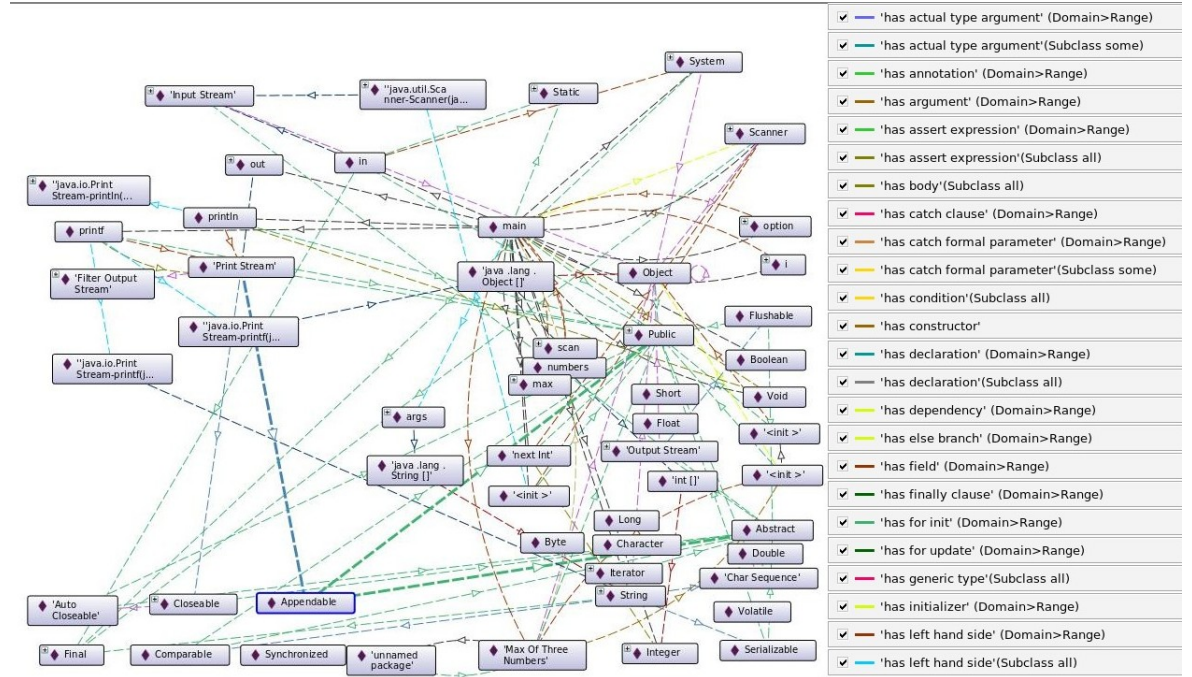


Figure 3.1: Topic map ontology created by source code written in 3.1 in JAVA programming language

3.2 Creating a Parser generator using ANTLR

ANTLR is a language recognition toolset which uses $LL(*)$ grammar for parsing. There are language grammars available including C ⁴, JavaScript ⁵, Java ⁶ etc. We used antlr to create a parse tree for the C source described in 2.1

Also, a parse tree for Javascript source from 2.2 was also created.

So, ANTLR can certainly recognize C and JS source files. By modifying antlr listener classes we can create toolset to generate dynamic rdf/xml files.

⁴<https://github.com/antlr/grammars-v4/tree/master/c>

⁵<https://github.com/antlr/grammars-v4/tree/master/ecmascript>

⁶<https://github.com/antlr/grammars-v4/tree/master/java>

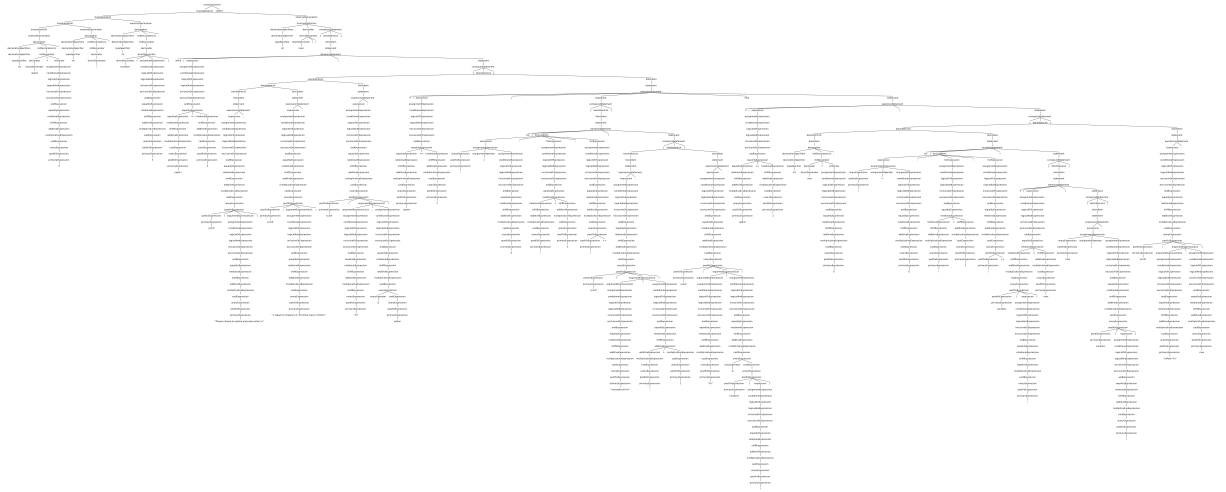


Figure 3.2: ANTLR parse tree created from C source at 2.1

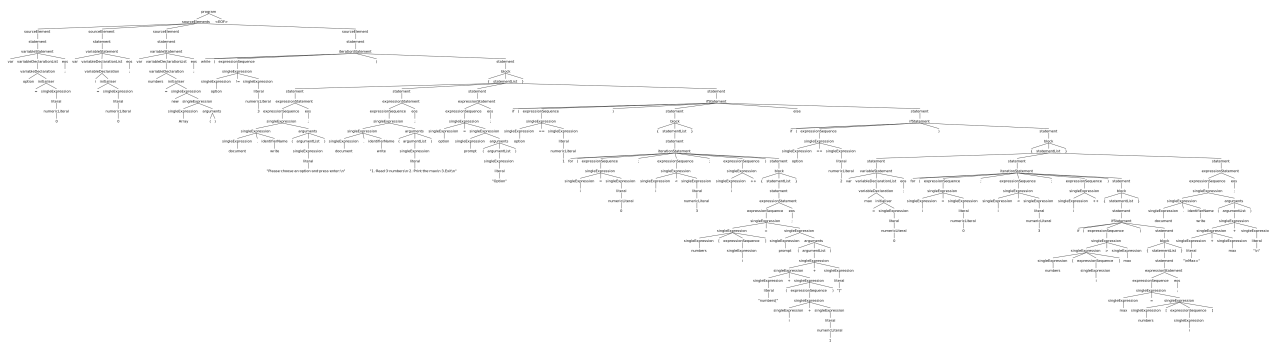


Figure 3.3: ANTLR parse tree created from JS source at 2.2

3.3 Modifying Flex/Bison Code Generators to create RDF/XML generators

Hypothetically, one can use the code generation techniques learned in CSE-310 course and modify the Assignment-4 (Code Generation) so that the parser generator outputs RDF/XML instead of ASSEMBLY code.

3.2 and 3.3 has not been fully implemented due to time shortage.

4

Appendix

4.1 Owl source code for C source code defined in 2.1

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns="https://semantic-web.netlify.com/
  ↪ c_source_ontology/"
3   xmlns:base="https://semantic-web.netlify.com/
  ↪ c_source_ontology/"
4   xmlns:owl="http://www.w3.org/2002/07/owl#"
5   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
6   xmlns:xml="http://www.w3.org/XML/1998/namespace"
7   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
8   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
9   xmlns:xkos="http://rdf-vocabulary.ddialliance.org/xkos#"
10  xmlns:dcterms="http://purl.org/dc/terms/"
11  xmlns:c_source_ontology="https://semantic-web.netlify.com/
  ↪ c_source_ontology/">
12  <owl:Ontology rdf:about="https://semantic-web.netlify.com/
  ↪ c_source_ontology/">
13    <rdfs:comment xml:lang="en">An ontology that represents
  ↪ structural and imperative programming languages</
  ↪ rdfs:comment>
14    <rdfs:label xml:lang="en">SoruceCodeOntology</
  ↪ rdfs:label>
15    </owl:Ontology>
16
17
18
19    <!--
20
  ↪ //////////////////////////////////////
  ↪
21    //
22    // Object Properties
23    //
```

```

24  ↪ //////////////////////////////////////
25  ↪
26  -->
27
28
29
30  <!-- https://semantic-web.netlify.com/c_source_ontology/
31  ↪ conditions -->
32
33  <owl:ObjectProperty rdf:about="https://semantic-web.netlify
34  ↪ .com/c_source_ontology/conditions">
35  ↪   <owl:inverseOf rdf:resource="https://semantic-web.
36  ↪ netlify.com/c_source_ontology/has_condition"/>
37  ↪   <rdfs:domain>
38  ↪     <owl:Restriction>
39  ↪       <owl:onProperty rdf:resource="https://semantic-
40  ↪ web.netlify.com/c_source_ontology/conditions"/>
41  ↪       <owl:someValuesFrom rdf:resource="https://
42  ↪ semantic-web.netlify.com/c_source_ontology/
43  ↪ ConditionalStructure"/>
44  ↪     </owl:Restriction>
45  ↪   </rdfs:domain>
46  ↪   <rdfs:range>
47  ↪     <owl:Restriction>
48  ↪       <owl:onProperty rdf:resource="https://semantic-
49  ↪ web.netlify.com/c_source_ontology/conditions"/>
50  ↪       <owl:someValuesFrom rdf:resource="https://
51  ↪ semantic-web.netlify.com/c_source_ontology/
52  ↪ RepetitiveStructure"/>
53  ↪     </owl:Restriction>
54  ↪   </rdfs:range>
55  ↪ </owl:ObjectProperty>
56
57  <!-- https://semantic-web.netlify.com/c_source_ontology/
58  ↪ has_condition -->
59
60  <owl:ObjectProperty rdf:about="https://semantic-web.netlify
61  ↪ .com/c_source_ontology/has_condition"/>
62
63  <!-- https://semantic-web.netlify.com/c_source_ontology/
64  ↪ has_type -->
65
66  <owl:ObjectProperty rdf:about="https://semantic-web.netlify
67  ↪ .com/c_source_ontology/has_type">
68  ↪   <owl:inverseOf rdf:resource="https://semantic-web.
69  ↪ netlify.com/c_source_ontology/is_type_of"/>
70  ↪   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
71  ↪ FunctionalProperty"/>

```

```

61     <rdfs:range rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/DataType"/>
62 </owl:ObjectProperty>
63
64
65
66 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ is_array_of -->
67
68 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/c_source_ontology/is_array_of">
69     <rdfs:domain rdf:resource="https://semantic-web.netlify
    ↪ .com/c_source_ontology/Variable"/>
70     <rdfs:range rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/DataType"/>
71 </owl:ObjectProperty>
72
73
74
75 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ is_else_branch_of -->
76
77 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/c_source_ontology/is_else_branch_of">
78     <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
    ↪ FunctionalProperty"/>
79     <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
    ↪ InverseFunctionalProperty"/>
80     <rdfs:comment>I have confusion about the
    ↪ Characteristics of the function.</rdfs:comment>
81 </owl:ObjectProperty>
82
83
84
85 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ is_included_in -->
86
87 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/c_source_ontology/is_included_in"/>
88
89
90
91 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ is_type_of -->
92
93 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/c_source_ontology/is_type_of"/>
94
95
96
97 <!--
98
    ↪ //////////////////////////////////////
    ↪

```

```

99      //
100     // Data properties
101     //
102
103     ↪ //////////////////////////////////////
104     ↪
105     -->
106
107
108     <!-- https://semantic-web.netlify.com/c_source_ontology/
109     ↪ dimension -->
110
111     <owl:DatatypeProperty rdf:about="https://semantic-web.
112     ↪ netlify.com/c_source_ontology/dimension">
113         <rdfs:subPropertyOf rdf:resource="http://www.w3.org
114     ↪ /2002/07/owl#topDataProperty"/>
115     </owl:DatatypeProperty>
116
117
118     <!-- https://semantic-web.netlify.com/c_source_ontology/
119     ↪ name -->
120
121     <owl:DatatypeProperty rdf:about="https://semantic-web.
122     ↪ netlify.com/c_source_ontology/name">
123         <rdfs:domain>
124             <owl:Class>
125                 <owl:unionOf rdf:parseType="Collection">
126                     <rdf:Description rdf:about="https://
127     ↪ semantic-web.netlify.com/c_source_ontology/Constants"/>
128                     <rdf:Description rdf:about="https://
129     ↪ semantic-web.netlify.com/c_source_ontology/DataType"/>
130                 </owl:unionOf>
131             </owl:Class>
132         </rdfs:domain>
133         <rdfs:range rdf:resource="http://www.w3.org/2001/
134     ↪ XMLSchema#string"/>
135     </owl:DatatypeProperty>
136
137
138     <!-- https://semantic-web.netlify.com/c_source_ontology/
139     ↪ source_code -->
140
141     <owl:DatatypeProperty rdf:about="https://semantic-web.
142     ↪ netlify.com/c_source_ontology/source_code">
143         <rdfs:domain>
144             <owl:Class>
145                 <owl:unionOf rdf:parseType="Collection">
146                     <rdf:Description rdf:about="https://
147     ↪ semantic-web.netlify.com/c_source_ontology/
148     ↪ ConditionalStructure"/>

```

```

139         <rdf:Description rdf:about="https://
↳ semantic-web.netlify.com/c_source_ontology/Constants"/>
140         <rdf:Description rdf:about="https://
↳ semantic-web.netlify.com/c_source_ontology/DataType"/>
141         <rdf:Description rdf:about="https://
↳ semantic-web.netlify.com/c_source_ontology/Operator"/>
142         <rdf:Description rdf:about="https://
↳ semantic-web.netlify.com/c_source_ontology/
↳ ProgrammingStructure"/>
143         <rdf:Description rdf:about="https://
↳ semantic-web.netlify.com/c_source_ontology/
↳ RepetitiveStructure"/>
144         <rdf:Description rdf:about="https://
↳ semantic-web.netlify.com/c_source_ontology/SystemFunction
↳ "/>
145         <rdf:Description rdf:about="https://
↳ semantic-web.netlify.com/c_source_ontology/Variable"/>
146         </owl:unionOf>
147         </owl:Class>
148         </rdfs:domain>
149         <rdfs:range rdf:resource="http://www.w3.org/2001/
↳ XMLSchema#string"/>
150         <rdfs:label xml:lang="en">source code</rdfs:label>
151     </owl:DatatypeProperty>
152
153
154
155     <!--
156
↳ //////////////////////////////////////
↳
157     //
158     // Classes
159     //
160
↳ //////////////////////////////////////
↳
161     -->
162
163
164
165
166     <!-- https://semantic-web.netlify.com/c_source_ontology/
↳ Array -->
167
168     <owl:Class rdf:about="https://semantic-web.netlify.com/
↳ c_source_ontology/Array">
169         <rdfs:subClassOf rdf:resource="https://semantic-web.
↳ netlify.com/c_source_ontology/DataType"/>
170     </owl:Class>
171
172
173

```

```

174 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↳ Comment -->
175
176 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↳ c_source_ontology/Comment"/>
177
178
179
180 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↳ ConditionalStructure -->
181
182 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↳ c_source_ontology/ConditionalStructure">
183     <rdfs:subClassOf rdf:resource="https://semantic-web.
    ↳ netlify.com/c_source_ontology/ProgrammingStructure"/>
184 </owl:Class>
185
186
187
188 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↳ Constants -->
189
190 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↳ c_source_ontology/Constants"/>
191
192
193
194 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↳ DataType -->
195
196 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↳ c_source_ontology/DataType"/>
197
198
199
200 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↳ Operator -->
201
202 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↳ c_source_ontology/Operator"/>
203
204
205
206 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↳ ProgrammingStructure -->
207
208 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↳ c_source_ontology/ProgrammingStructure"/>
209
210
211
212 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↳ RepetitiveStructure -->
213

```



```

214 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↪ c_source_ontology/RepetitiveStructure">
215 <rdfs:subClassOf rdf:resource="https://semantic-web.
    ↪ netlify.com/c_source_ontology/ProgrammingStructure"/>
216 </owl:Class>
217
218
219
220 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ SystemFunction -->
221
222 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↪ c_source_ontology/SystemFunction"/>
223
224
225
226 <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ Variable -->
227
228 <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↪ c_source_ontology/Variable"/>
229
230
231
232 <!--
233
    ↪ //////////////////////////////////////
    ↪
234 //
235 // Individuals
236 //
237
    ↪ //////////////////////////////////////
    ↪
238 -->
239
240
241
242
243 <!-- https://semantic-web.netlify.com/c_source_ontology/For
    ↪ -->
244
245 <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/For">
246 <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/RepetitiveStructure"/>
247 <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/c_source_ontology/if"/>
248 <source_code rdf:datatype="http://www.w3.org/2001/
    ↪/XMLSchema#string">for (i=0; i<3; i++) {
249     printf("&quot;\nnumbers[%i]=&quot;;,i+1); scanf("&quot;;%i&
    ↪ quot;;,&amp;numbers[i]);
250 }</source_code>
251 </owl:NamedIndividual>

```

```

252
253
254
255 <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ For2 -->
256
257 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/For2">
258   <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/RepetitiveStructure"/>
259   <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/else_if"/>
260   <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">for (i=0; i<3; i++) {
261     if(numbers[i] > max) {
262       max = numbers[i];
263     }
264   }</source_code>
265 </owl:NamedIndividual>
266
267
268
269 <!-- https://semantic-web.netlify.com/c_source_ontology/Int
↪ -->
270
271 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/Int">
272   <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/DataType"/>
273   <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">int</name>
274   <rdfs:comment xml:lang="en">The int type</rdfs:comment>
275   <rdfs:label xml:lang="en">Int</rdfs:label>
276 </owl:NamedIndividual>
277
278
279
280 <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ Void -->
281
282 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/Void">
283   <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/DataType"/>
284   <rdfs:label rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">void</rdfs:label>
285 </owl:NamedIndividual>
286
287
288
289 <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ While -->
290

```

```

291 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/While">
292 <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/RepetitiveStructure"/>
293 <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">while</name>
294 <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">while (option!=3){
295 printf(&quot;Please choose an option and press enter:\n&
↪ quot;);
296 printf(&quot;1. Read 3 numbers\n 2. Print the max\n 3.Exit\
↪ n&quot;);
297 scanf(&quot;%i&quot;;,&amp;option);
298 if (option==1) {
299     for (i=0; i<3; i++) {
300         printf(&quot;\nnumbers [%i]=&quot;;,i+1);scanf(&quot;
↪ %i&quot;;,&amp;numbers[i]);
301     }
302 } else if (option==2) {
303     int max = 0;
304     for (i=0; i<3; i++) {
305         if(numbers[i] > max) {
306             max = numbers[i];
307         }
308     }
309     printf(&quot;\nMax=%i&quot;;,max);
310 }
311 }</source_code>
312 </owl:NamedIndividual>
313
314
315
316 <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ else_if -->
317
318 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/else_if">
319 <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/ConditionalStructure"/>
320 <is_else_branch_of rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/if"/>
321 <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/While"/>
322 <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">else if (option==2) {
323     int max = 0;
324     for (i=0; i<3; i++) {
325         if(numbers[i] > max) {
326             max = numbers[i];
327         }
328     }
329     printf(&quot;\nMax=%i&quot;;,max);
330 }</source_code>
331 </owl:NamedIndividual>

```

```

332
333
334
335 <!-- https://semantic-web.netlify.com/c_source_ontology/i
↪ -->
336
337 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/i">
338 <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Variable"/>
339 <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/While"/>
340 <is_type_of rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Int"/>
341 <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">i</name>
342 <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">int i</source_code>
343 <rdfs:label rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">i</rdfs:label>
344 </owl:NamedIndividual>
345
346
347
348 <!-- https://semantic-web.netlify.com/c_source_ontology/if
↪ -->
349
350 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/if">
351 <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/ConditionalStructure"/>
352 <is_else_branch_of rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/else_if"/>
353 <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/While"/>
354 <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">if (option==1) {
355     for (i=0; i<3; i++) {
356         printf(&quot;\nnumbers [%i]=&quot;;,i+1); scanf(&quot;
↪ %i&quot;;,&amp;numbers[i]);
357     }
358 }</source_code>
359 <rdfs:label rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">if</rdfs:label>
360 </owl:NamedIndividual>
361
362
363
364 <!-- https://semantic-web.netlify.com/c_source_ontology/if2
↪ -->
365
366 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/if2">

```

```

367     <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/ConditionalStructure"/>
368     <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/For2"/>
369     <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">if(numbers[i] > max) {
370         max = numbers[i];
371     }</source_code>
372     </owl:NamedIndividual>
373
374
375
376     <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ int_array -->
377
378     <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/int_array">
379         <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Array"/>
380         <dimension rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#int">1</dimension>
381         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">integer array</name>
382         <rdfs:label rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">Integer array</rdfs:label>
383     </owl:NamedIndividual>
384
385
386
387     <!-- https://semantic-web.netlify.com/c_source_ontology/max
↪ -->
388
389     <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/max">
390         <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Variable"/>
391         <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/else_if"/>
392     </owl:NamedIndividual>
393
394
395
396     <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ numbers -->
397
398     <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/numbers">
399         <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Variable"/>
400         <has_type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/int_array"/>
401         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">numbers</name>

```

```

402     <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">int numbers[3]</source_code>
403 </owl:NamedIndividual>
404
405
406
407 <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ option -->
408
409 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/option">
410     <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Variable"/>
411     <is_type_of rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Int"/>
412     <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">option</name>
413     <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">int option = 0</source_code>
414     <rdfs:label rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">option</rdfs:label>
415 </owl:NamedIndividual>
416
417
418
419 <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ printf -->
420
421 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/printf">
422     <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/SystemFunction"/>
423     <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/While"/>
424     <is_type_of rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/Void"/>
425     <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">printf</name>
426     <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">printf(&quot;Please choose an option
↪ and press enter:\n&quot;);</source_code>
427 </owl:NamedIndividual>
428
429
430
431 <!-- https://semantic-web.netlify.com/c_source_ontology/
↪ printf2 -->
432
433 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/c_source_ontology/printf2">
434     <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/c_source_ontology/SystemFunction"/>
435     <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/c_source_ontology/While"/>

```

```

436         <is_type_of rdf:resource="https://semantic-web.netlify.
↳ com/c_source_ontology/Void"/>
437         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↳ string">printf</name>
438         <source_code rdf:datatype="http://www.w3.org/2001/
↳ XMLSchema#string">printf(&quot;1. Read 3 numbers\n 2.
↳ Print the max\n 3.Exit\n&quot;);</source_code>
439     </owl:NamedIndividual>
440
441
442
443     <!-- https://semantic-web.netlify.com/c_source_ontology/
↳ printf3 -->
444
445     <owl:NamedIndividual rdf:about="https://semantic-web.
↳ netlify.com/c_source_ontology/printf3">
446         <rdf:type rdf:resource="https://semantic-web.netlify.
↳ com/c_source_ontology/SystemFunction"/>
447         <is_included_in rdf:resource="https://semantic-web.
↳ netlify.com/c_source_ontology/For"/>
448         <is_type_of rdf:resource="https://semantic-web.netlify.
↳ com/c_source_ontology/Void"/>
449         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↳ string">printf</name>
450         <source_code rdf:datatype="http://www.w3.org/2001/
↳ XMLSchema#string">printf(&quot;\nnumbers [%i]=&quot;;i+1);
↳ scanf(&quot;%i&quot;;,&amp;numbers[i]);</source_code>
451     </owl:NamedIndividual>
452
453
454
455     <!-- https://semantic-web.netlify.com/c_source_ontology/
↳ printf4 -->
456
457     <owl:NamedIndividual rdf:about="https://semantic-web.
↳ netlify.com/c_source_ontology/printf4">
458         <rdf:type rdf:resource="https://semantic-web.netlify.
↳ com/c_source_ontology/SystemFunction"/>
459         <is_included_in rdf:resource="https://semantic-web.
↳ netlify.com/c_source_ontology/else_if"/>
460         <is_type_of rdf:resource="https://semantic-web.netlify.
↳ com/c_source_ontology/Void"/>
461         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↳ string">printf</name>
462         <source_code rdf:datatype="http://www.w3.org/2001/
↳ XMLSchema#string">printf(&quot;\nMax=%i&quot;;,max);</
↳ source_code>
463     </owl:NamedIndividual>
464
465
466
467     <!-- https://semantic-web.netlify.com/c_source_ontology/
↳ scanf -->
468

```

```

469     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/scanf">
470         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/SystemFunction"/>
471         <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/c_source_ontology/While"/>
472         <is_type_of rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/Void"/>
473         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">scanf</name>
474         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">scanf(&quot;%i&quot;;,&amp;option);</
    ↪ source_code>
475         <rdfs:label rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">scanf</rdfs:label>
476     </owl:NamedIndividual>
477 </rdf:RDF>
478
479
480
481 <!-- Generated by the OWL API (version 4.5.9.2019-02-01
    ↪ T07:24:44Z) https://github.com/owlcs/owlapi -->

```

Listing 4.1: Owl source code for C source code defined in 2.1

4.2 Owl source code for JavaScript source code defined in 2.2

```

1 <?xml version="1.0"?>
2 <rdf:RDF xmlns="https://semantic-web.netlify.com/
    ↪ js_source_ontology/"
3     xml:base="https://semantic-web.netlify.com/
    ↪ js_source_ontology/"
4     xmlns:owl="http://www.w3.org/2002/07/owl#"
5     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
6     xmlns:xml="http://www.w3.org/XML/1998/namespace"
7     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
8     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
9     xmlns:xkos="http://rdf-vocabulary.ddialliance.org/xkos#"
10    xmlns:dcterms="http://purl.org/dc/terms/"
11    xmlns:c_source_ontology="https://semantic-web.netlify.com/
    ↪ c_source_ontology/">
12    <owl:Ontology rdf:about="https://semantic-web.netlify.com/
    ↪ js_source_ontology/">
13        <rdfs:comment xml:lang="en">An ontology that represents
    ↪ structural and imperative programming languages</
    ↪ rdfs:comment>
14        <rdfs:label xml:lang="en">CodeOntology_Modified</
    ↪ rdfs:label>
15    </owl:Ontology>
16
17
18

```



```

19 <!--
20
21 //
22 // Object Properties
23 //
24
25 -->
26
27
28
29
30 <!-- https://semantic-web.netlify.com/js_source_ontology/
31 conditions -->
32
33 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
34 .com/js_source_ontology/conditions">
35   <owl:inverseOf rdf:resource="https://semantic-web.
36   netlify.com/js_source_ontology/has_condition"/>
37   <rdfs:domain>
38     <owl:Restriction>
39       <owl:onProperty rdf:resource="https://semantic-
40       web.netlify.com/js_source_ontology/conditions"/>
41       <owl:someValuesFrom rdf:resource="https://
42       semantic-web.netlify.com/js_source_ontology/
43       ConditionalStructure"/>
44     </owl:Restriction>
45   </rdfs:domain>
46   <rdfs:range>
47     <owl:Restriction>
48       <owl:onProperty rdf:resource="https://semantic-
49       web.netlify.com/js_source_ontology/conditions"/>
50       <owl:someValuesFrom rdf:resource="https://
51       semantic-web.netlify.com/js_source_ontology/
52       RepetitiveStructure"/>
53     </owl:Restriction>
54   </rdfs:range>
55 </owl:ObjectProperty>
56
57 <!-- https://semantic-web.netlify.com/js_source_ontology/
58 has_condition -->
59
60 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
61 .com/js_source_ontology/has_condition"/>
62
63
64 <!-- https://semantic-web.netlify.com/js_source_ontology/
65 has_type -->

```

```

57
58 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
↪ .com/js_source_ontology/has_type">
59 <owl:inverseOf rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/is_type_of"/>
60 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
↪ FunctionalProperty"/>
61 <rdfs:range rdf:resource="https://semantic-web.netlify.
↪ com/js_source_ontology/DataType"/>
62 </owl:ObjectProperty>
63
64
65
66 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ is_array_of -->
67
68 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
↪ .com/js_source_ontology/is_array_of">
69 <rdfs:domain rdf:resource="https://semantic-web.netlify
↪ .com/js_source_ontology/Variable"/>
70 <rdfs:range rdf:resource="https://semantic-web.netlify.
↪ com/js_source_ontology/DataType"/>
71 </owl:ObjectProperty>
72
73
74
75 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ is_else_branch_of -->
76
77 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
↪ .com/js_source_ontology/is_else_branch_of">
78 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
↪ FunctionalProperty"/>
79 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
↪ InverseFunctionalProperty"/>
80 <rdfs:comment>I have confusion about the
↪ Characteristics of the function.</rdfs:comment>
81 </owl:ObjectProperty>
82
83
84
85 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ is_included_in -->
86
87 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
↪ .com/js_source_ontology/is_included_in"/>
88
89
90
91 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ is_type_of -->
92
93 <owl:ObjectProperty rdf:about="https://semantic-web.netlify
↪ .com/js_source_ontology/is_type_of"/>

```

```

94
95
96
97 <!--
98
99 //
100 // Data properties
101 //
102
103 -->
104
105
106
107
108 <!-- https://semantic-web.netlify.com/js_source_ontology/
109 dimension -->
110
111 <owl:DatatypeProperty rdf:about="https://semantic-web.
112 netlify.com/js_source_ontology/dimension">
113   <rdfs:subPropertyOf rdf:resource="http://www.w3.org
114   /2002/07/owl#topDataProperty"/>
115 </owl:DatatypeProperty>
116
117 <!-- https://semantic-web.netlify.com/js_source_ontology/
118 name -->
119
120 <owl:DatatypeProperty rdf:about="https://semantic-web.
121 netlify.com/js_source_ontology/name">
122   <rdfs:domain>
123     <owl:Class>
124       <owl:unionOf rdf:parseType="Collection">
125         <rdf:Description rdf:about="https://
126 semantic-web.netlify.com/js_source_ontology/Constants"/>
127         <rdf:Description rdf:about="https://
128 semantic-web.netlify.com/js_source_ontology/DataType"/>
129       </owl:unionOf>
130     </owl:Class>
131   </rdfs:domain>
132   <rdfs:range rdf:resource="http://www.w3.org/2001/
133 XMLSchema#string"/>
134 </owl:DatatypeProperty>

```

```

135     ↪ netlify.com/js_source_ontology/source_code">
136         <rdfs:domain>
137             <owl:Class>
138                 <owl:unionOf rdf:parseType="Collection">
139                     <rdf:Description rdf:about="https://
140 ↪ semantic-web.netlify.com/js_source_ontology/
141 ↪ ConditionalStructure"/>
142                     <rdf:Description rdf:about="https://
143 ↪ semantic-web.netlify.com/js_source_ontology/Constants"/>
144                     <rdf:Description rdf:about="https://
145 ↪ semantic-web.netlify.com/js_source_ontology/DataType"/>
146                     <rdf:Description rdf:about="https://
147 ↪ semantic-web.netlify.com/js_source_ontology/Operator"/>
148                     <rdf:Description rdf:about="https://
149 ↪ semantic-web.netlify.com/js_source_ontology/
150 ↪ ProgrammingStructure"/>
151                     <rdf:Description rdf:about="https://
152 ↪ semantic-web.netlify.com/js_source_ontology/
153 ↪ RepetitiveStructure"/>
154                     <rdf:Description rdf:about="https://
155 ↪ semantic-web.netlify.com/js_source_ontology/
156 ↪ SystemFunction"/>
157                     <rdf:Description rdf:about="https://
158 ↪ semantic-web.netlify.com/js_source_ontology/Variable"/>
159                 </owl:unionOf>
160             </owl:Class>
161         </rdfs:domain>
162         <rdfs:range rdf:resource="http://www.w3.org/2001/
163 ↪ XMLSchema#string"/>
164         <rdfs:label xml:lang="en">source code</rdfs:label>
165     </owl:DatatypeProperty>
166
167     <!--
168     ↪ //////////////////////////////////////
169     ↪
170     //
171     // Classes
172     //
173     ↪ //////////////////////////////////////
174     ↪
175     -->
176
177     <!-- https://semantic-web.netlify.com/js_source_ontology/
178     ↪ Array -->
179
180     <owl:Class rdf:about="https://semantic-web.netlify.com/
181     ↪ js_source_ontology/Array">

```

```

169         <rdfs:subClassOf rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/DataType"/>
170     </owl:Class>
171
172
173
174     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ Comment -->
175
176     <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/Comment"/>
177
178
179
180     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ ConditionalStructure -->
181
182     <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/ConditionalStructure">
183         <rdfs:subClassOf rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/ProgrammingStructure"/>
184     </owl:Class>
185
186
187
188     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ Constants -->
189
190     <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/Constants"/>
191
192
193
194     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ DataType -->
195
196     <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/DataType"/>
197
198
199
200     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ Operator -->
201
202     <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/Operator"/>
203
204
205
206     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ ProgrammingStructure -->
207
208     <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/ProgrammingStructure"/>

```

```

209
210
211
212 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ RepetitiveStructure -->
213
214 <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/RepetitiveStructure">
215 <rdfs:subClassOf rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/ProgrammingStructure"/>
216 </owl:Class>
217
218
219
220 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ SystemFunction -->
221
222 <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/SystemFunction"/>
223
224
225
226 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ Variable -->
227
228 <owl:Class rdf:about="https://semantic-web.netlify.com/
↪ js_source_ontology/Variable"/>
229
230
231
232 <!--
233
↪ //////////////////////////////////////
↪
234 //
235 // Individuals
236 //
237
↪ //////////////////////////////////////
↪
238 -->
239
240
241
242
243 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ For -->
244
245 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/js_source_ontology/For">
246 <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/js_source_ontology/RepetitiveStructure"/>
247 <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/if"/>

```

```

248     <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">for (i=0; i<3; i++) {
249         document.write("&quot;\nnumbers[%i]=&quot;;,i+1);scanf(&
    ↪ quot;%i&quot;;,&amp;numbers[i]);
250 }</source_code>
251     </owl:NamedIndividual>
252
253
254
255     <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ For2 -->
256
257     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/For2">
258         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/RepetitiveStructure"/>
259         <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/else_if"/>
260         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">for (i=0; i<3; i++) {
261             if(numbers[i] &gt; max) {
262                 max = numbers[i];
263             }
264 }</source_code>
265     </owl:NamedIndividual>
266
267
268
269     <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ Void -->
270
271     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/Void">
272         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/DataType"/>
273         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">void</name>
274         <rdfs:label rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">void</rdfs:label>
275     </owl:NamedIndividual>
276
277
278
279     <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ While -->
280
281     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/While">
282         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/RepetitiveStructure"/>
283         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">while</name>
284         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">while (option!=3){

```

```

285     document.write("&quot;Please choose an option and press
↪ enter:\n&quot;);
286     document.write("&quot;1. Read 3 numbers\n 2. Print the max\n
↪ 3.Exit\n&quot;);
287     option = prompt("&quot;Option&quot;");
288     if (option == 1) {
289         for (i=0; i<3; i++) {
290             numbers[i] = prompt("&quot;numbers&quot; + (i+1) +
↪ &quot;]&quot;);
291         }
292     } else if (option == 2) {
293         var max = 0;
294         for (i=0; i<3; i++) {
295             if(numbers[i] > max) {
296                 max = numbers[i];
297             }
298         }
299         document.write("&quot;\nMax=&quot; + max + &quot;\n&quot;
↪ );
300     }
301 }</source_code>
302 </owl:NamedIndividual>
303
304
305
306 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ array -->
307
308 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/js_source_ontology/array">
309     <rdf:type rdf:resource="https://semantic-web.
↪ com/js_source_ontology/Array"/>
310     <dimension rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#int">1</dimension>
311     <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">array</name>
312     <rdfs:label rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">Integer array</rdfs:label>
313 </owl:NamedIndividual>
314
315
316
317 <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ else_if -->
318
319 <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/js_source_ontology/else_if">
320     <rdf:type rdf:resource="https://semantic-web.
↪ com/js_source_ontology/ConditionalStructure"/>
321     <is_else_branch_of rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/if"/>
322     <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/While"/>
323     <source_code rdf:datatype="http://www.w3.org/2001/

```



```

324     ↪ XMLSchema#string">else if (option==2) {
325         var max = 0;
326         for (i=0; i<3; i++) {
327             if(numbers[i] > max) {
328                 max = numbers[i];
329             }
330         }
331         document.write("&quot;\nMax=%i&quot;;max);
332     }</source_code>
333     </owl:NamedIndividual>
334
335
336     <!-- https://semantic-web.netlify.com/js_source_ontology/i
337     ↪ -->
338
339     <owl:NamedIndividual rdf:about="https://semantic-web.
340     ↪ netlify.com/js_source_ontology/i">
341         <rdf:type rdf:resource="https://semantic-web.
342     ↪ com/js_source_ontology/Variable"/>
343         <is_included_in rdf:resource="https://semantic-web.
344     ↪ netlify.com/js_source_ontology/While"/>
345         <is_type_of rdf:resource="https://semantic-web.
346     ↪ com/js_source_ontology/var"/>
347         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
348     ↪ string">i</name>
349         <source_code rdf:datatype="http://www.w3.org/2001/
350     ↪ XMLSchema#string">int i</source_code>
351         <rdfs:label rdf:datatype="http://www.w3.org/2001/
352     ↪ XMLSchema#string">i</rdfs:label>
353     </owl:NamedIndividual>
354
355
356     <!-- https://semantic-web.netlify.com/js_source_ontology/if
357     ↪ -->
358
359     <owl:NamedIndividual rdf:about="https://semantic-web.
360     ↪ netlify.com/js_source_ontology/if">
361         <rdf:type rdf:resource="https://semantic-web.
362     ↪ com/js_source_ontology/ConditionalStructure"/>
363         <is_else_branch_of rdf:resource="https://semantic-web.
364     ↪ netlify.com/js_source_ontology/else_if"/>
365         <is_included_in rdf:resource="https://semantic-web.
366     ↪ netlify.com/js_source_ontology/While"/>
367         <source_code rdf:datatype="http://www.w3.org/2001/
368     ↪ XMLSchema#string">if (option==1) {
369             for (i=0; i<3; i++) {
370                 document.write("&quot;\nnumbers[%i]=&quot;;i+1);
371     ↪ scanf("&quot;%i&quot;;,&amp;numbers[i]);
372             }
373         }</source_code>
374         <rdfs:label rdf:datatype="http://www.w3.org/2001/
375     ↪ XMLSchema#string">if</rdfs:label>

```

```

361     </owl:NamedIndividual>
362
363
364
365     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ if2 -->
366
367     <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/js_source_ontology/if2">
368         <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/js_source_ontology/ConditionalStructure"/>
369         <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/For2"/>
370         <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">if(numbers[i] &gt; max) {
371             max = numbers[i];
372         }</source_code>
373     </owl:NamedIndividual>
374
375
376
377     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ max -->
378
379     <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/js_source_ontology/max">
380         <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/js_source_ontology/Variable"/>
381         <is_included_in rdf:resource="https://semantic-web.
↪ netlify.com/js_source_ontology/else_if"/>
382         <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">var max = 0</source_code>
383     </owl:NamedIndividual>
384
385
386
387     <!-- https://semantic-web.netlify.com/js_source_ontology/
↪ numbers -->
388
389     <owl:NamedIndividual rdf:about="https://semantic-web.
↪ netlify.com/js_source_ontology/numbers">
390         <rdf:type rdf:resource="https://semantic-web.netlify.
↪ com/js_source_ontology/Variable"/>
391         <has_type rdf:resource="https://semantic-web.netlify.
↪ com/js_source_ontology/array"/>
392         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
↪ string">numbers</name>
393         <source_code rdf:datatype="http://www.w3.org/2001/
↪ XMLSchema#string">int numbers[3]</source_code>
394     </owl:NamedIndividual>
395
396
397
398     <!-- https://semantic-web.netlify.com/js_source_ontology/

```

```

399 ↪ option -->
400
401 <owl:NamedIndividual rdf:about="https://semantic-web.
402 ↪ netlify.com/js_source_ontology/option">
403 <rdf:type rdf:resource="https://semantic-web.netlify.
404 ↪ com/js_source_ontology/Variable"/>
405 <is_type_of rdf:resource="https://semantic-web.netlify.
406 ↪ com/js_source_ontology/var"/>
407 <source_code rdf:datatype="http://www.w3.org/2001/
408 ↪ XMLSchema#string">var option = 0;</source_code>
409 <rdfs:label rdf:datatype="http://www.w3.org/2001/
410 ↪ XMLSchema#string">option</rdfs:label>
411 </owl:NamedIndividual>
412
413 <!-- https://semantic-web.netlify.com/js_source_ontology/
414 ↪ prompt -->
415
416 <owl:NamedIndividual rdf:about="https://semantic-web.
417 ↪ netlify.com/js_source_ontology/prompt">
418 <rdf:type rdf:resource="https://semantic-web.netlify.
419 ↪ com/js_source_ontology/SystemFunction"/>
420 <is_included_in rdf:resource="https://semantic-web.
421 ↪ netlify.com/js_source_ontology/While"/>
422 <is_type_of rdf:resource="https://semantic-web.netlify.
423 ↪ com/js_source_ontology/Void"/>
424 <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
425 ↪ string">scanf</name>
426 <source_code rdf:datatype="http://www.w3.org/2001/
427 ↪ XMLSchema#string">option = prompt(&quot;Option&quot;)</
428 ↪ source_code>
429 <rdfs:label rdf:datatype="http://www.w3.org/2001/
430 ↪ XMLSchema#string">prompt</rdfs:label>
431 </owl:NamedIndividual>
432
433 <!-- https://semantic-web.netlify.com/js_source_ontology/
434 ↪ var -->
435
436 <owl:NamedIndividual rdf:about="https://semantic-web.
437 ↪ netlify.com/js_source_ontology/var">
438 <rdf:type rdf:resource="https://semantic-web.netlify.
439 ↪ com/js_source_ontology/DataType"/>
440 <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
441 ↪ string">var</name>
442 <rdfs:comment xml:lang="en">Javascript dynamic datatype
443 ↪ </rdfs:comment>
444 <rdfs:label xml:lang="en">var</rdfs:label>
445 </owl:NamedIndividual>

```

```

433 <!-- https://semantic-web.netlify.com/js_source_ontology/
434 ↪ write -->
435
436 <owl:NamedIndividual rdf:about="https://semantic-web.
437 ↪ netlify.com/js_source_ontology/write">
438 <rdf:type rdf:resource="https://semantic-web.netlify.
439 ↪ com/js_source_ontology/SystemFunction"/>
440 <is_included_in rdf:resource="https://semantic-web.
441 ↪ netlify.com/js_source_ontology/While"/>
442 <is_type_of rdf:resource="https://semantic-web.netlify.
443 ↪ com/js_source_ontology/Void"/>
444 <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
445 ↪ string">document.write</name>
446 <source_code rdf:datatype="http://www.w3.org/2001/
447 ↪ XMLSchema#string">document.write(&quot;Please choose an
448 ↪ option and press enter:\n&quot;);</source_code>
449 </owl:NamedIndividual>
450
451 <!-- https://semantic-web.netlify.com/js_source_ontology/
452 ↪ write2 -->
453
454 <owl:NamedIndividual rdf:about="https://semantic-web.
455 ↪ netlify.com/js_source_ontology/write2">
456 <rdf:type rdf:resource="https://semantic-web.netlify.
457 ↪ com/js_source_ontology/SystemFunction"/>
458 <is_included_in rdf:resource="https://semantic-web.
459 ↪ netlify.com/js_source_ontology/While"/>
460 <is_type_of rdf:resource="https://semantic-web.netlify.
461 ↪ com/js_source_ontology/Void"/>
462 <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
463 ↪ string">document.write</name>
464 <source_code rdf:datatype="http://www.w3.org/2001/

```

```

465     ↪ XMLSchema#string">document.write("&quot;\nMax= &quot; +
466     ↪ max + &quot;\n&quot;);</source_code>
467     </owl:NamedIndividual>
468 </rdf:RDF>
469
470 <!-- Generated by the OWL API (version 4.5.9.2019-02-01
471     ↪ T07:24:44Z) https://github.com/owlcs/owlapi -->

```

Listing 4.2: Owl source code for JS source code defined in 2.2