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

List of Figures

Listings

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.

```
#include <stdio.h>

int option = 0;
int i;
int numbers[3];

int main (){
    while (option!=3){
        printf("Please choose an option and press enter:\n");
        printf("1. Read 3 numbers\n 2. Print the max\n 3.Exit\n
    ↪ ");
        scanf("%i",&option);
        if (option==1) {
            for (i=0; i<3; i++) {
                printf("\nnumbers[%i]=",i+1);
                scanf("%i",&numbers[i]);
            }
```

```
        } else if (option==2) {
            int max = 0;
            for (i=0; i<3; i++) {
                if(numbers[i] > max) {
                    max = numbers[i];
                }
            }
            printf("\nMax=%i",max);
        }
    }
}
```

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

And another is in the Javascript programming language.

```
var option = 0;
var i = 0;
var numbers = new Array();

while (option!=3){
    document.write("Please choose an option and press enter:\n"
    ↪ );
    document.write("1. Read 3 numbers\n 2. Print the max\n 3.
    ↪ Exit\n");
    option = prompt("Option");
    if (option == 1) {
        for (i=0; i<3; i++) {
            numbers[i] = prompt("numbers[" + (i+1) + "]");
        }
    } else if (option == 2) {
        var max = 0;
        for (i=0; i<3; i++) {
            if(numbers[i] > max) {
                max = numbers[i];
            }
        }
        document.write("\nMax=" + max + "\n");
    }
}
```

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. it We get ontology 4.1for 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.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

# For C source code
BASE <https://semantic-web.netlify.com/c_source_ontology/>

# For Javascript Code
BASE <https://semantic-web.netlify.com/js_source_ontology/>


# 1. Total number of conditional structures

SELECT (COUNT(?subject) AS ?c) WHERE {
    ?subject rdf:type <ConditionalStructure>
}

# Expected Result on C code: 3
# Expected Result on JS code: 3

# 2. Total number of repetitive structures

SELECT (COUNT(?subject) AS ?c) WHERE {
    ?subject rdf:type <RepetitiveStructure>
}

# Expected Result on C code: 3
# Expected Result on JS code: 3

# 3. Total number of variables

SELECT (COUNT(?subject) AS ?c) WHERE {
    ?subject rdf:type <Variable>
}


```

[1] https://jena.apache.org/

```
# Expected Result on C code: 4
# Expected Result on JS code: 4

# 4. Total number of conditional structures included in
    ↪ repetitive structures


SELECT (COUNT(?subject) AS ?c) WHERE {
    ?object rdf:type <RepetitiveStructure> .
    ?subject <is_included_in> ?object .
    ?subject rdf:type <ConditionalStructure>
}

# Expected Result on C code: 3
# Expected Result on JS code: 3

# 5. Total number of repetitive structures included in
    ↪ repetitive structures

SELECT (COUNT(?subject) AS ?c) WHERE {
    ?object rdf:type <RepetitiveStructure> .
    ?subject <is_included_in> ?object .
    ?subject rdf:type <RepetitiveStructure>
}

# Expected Result on C code: 0
# Expected Result on JS code: 0

# 6.  Total number of system functions called

SELECT (COUNT(?subject) AS ?c) WHERE {
    ?subject rdf:type <SystemFunction>
}

# Expected Result on C code: 5
# Expected Result on JS code: 4

# 7.  Total number of system functions called in conditional
    ↪ structures

SELECT (COUNT(?subject) AS ?c) WHERE {
    ?object rdf:type <ConditionalStructure> .
    ?subject <is_included_in> ?object .
    ?subject rdf:type <SystemFunction>
 }

 # Expected Result on C code: 1
 # Expected Result on JS code: 1

# 8. Total Number of repetative structures

SELECT (COUNT(?subject) AS ?c) WHERE {
    ?object rdf:type <RepetitiveStructure> .
    ?subject <is_included_in> ?object .
```

```
    ?subject rdf:type <SystemFunction>
}

# Expected Result on C code: 4
# 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 folling map of the C source ontology at 4.1 found from source 2.1.

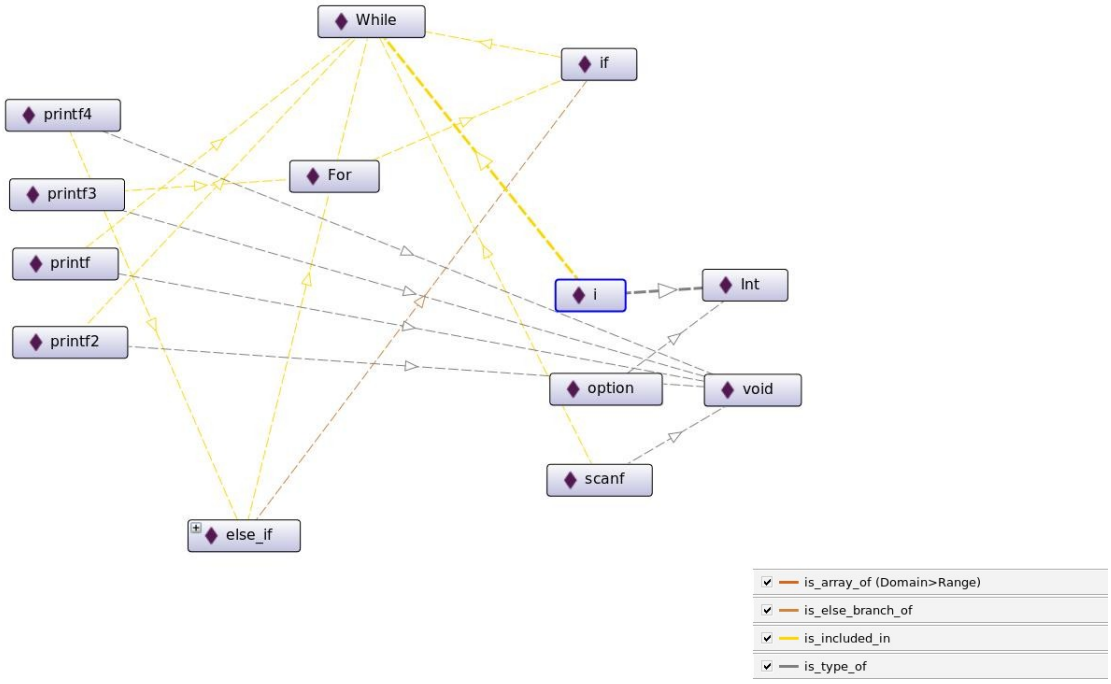

Figure 2.1: Topic map ontology created by source code written in 2.1 in C language

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

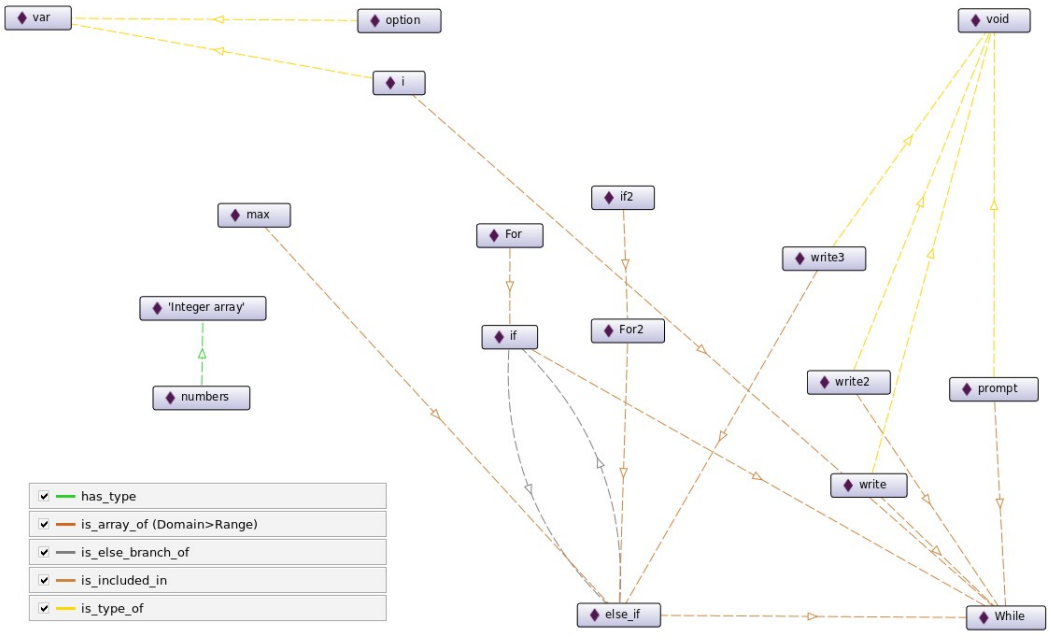

Figure 2.2: Topic map ontology created by source code written in 2.2 in JS

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-1 [1] [2] already provides a RDF triple generating parser [3]. 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

```
import java.util.Scanner;
public class MaxOfThreeNumbers {
    public static void main(String[] args) {
        int option = 0;
        int i;
        int[] numbers = new int[3];

        while (option!=3){
            System.out.println("Please choose an option and
    ↪ press enter:\n");
            System.out.println("1. Read 3 numbers\n 2. Print
    ↪ the max\n 3.Exit\n");

            Scanner scan = new Scanner(System.in);
            option = scan.nextInt();

            if (option==1) {
                for (i=0; i<3; i++) {
                    System.out.printf("\nnumbers[%d]=",i+1);
                    numbers[i] = scan.nextInt();
                }
            } else if (option==2) {
                int max = 0;
                for (i=0; i<3; i++) {
                    if(numbers[i] > max) {
                        max = numbers[i];
                    }
                }
                System.out.printf("\nMax=%d",max);
            }
        }


    }
}
```

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

[1] CodeOntology: 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.

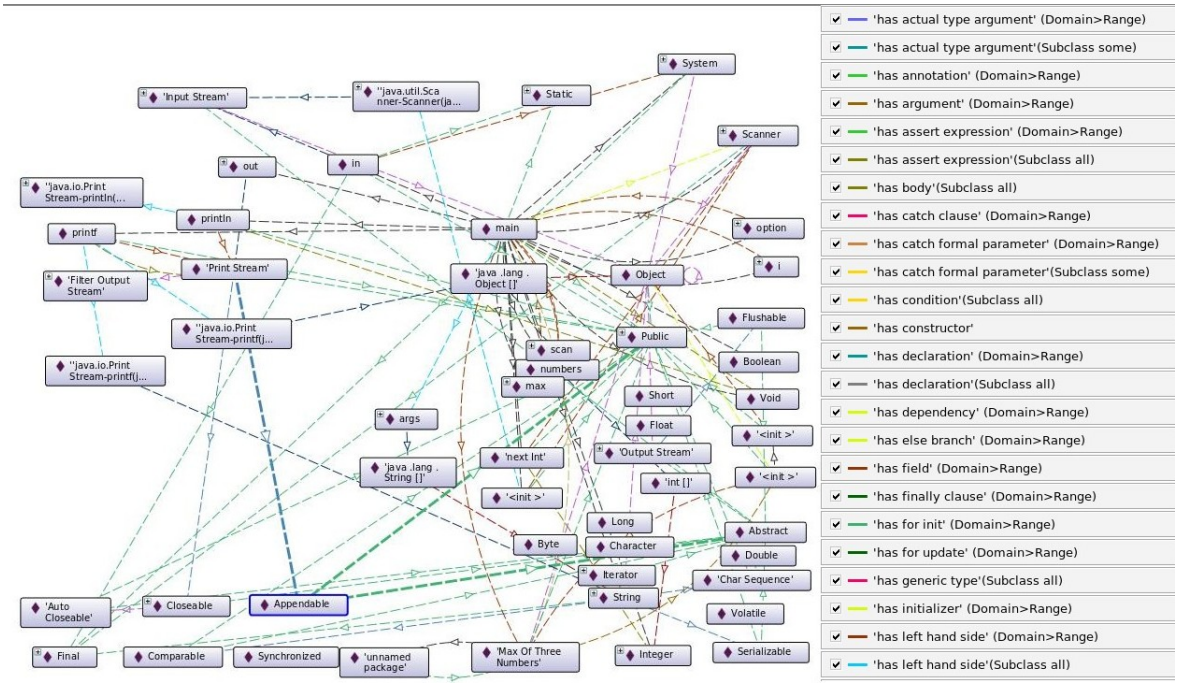

Figure 3.1: Topic map ontology created by source code written in 3.1 in JAVA programming language

3.2 Creating a Parser generator using ANTLR

ANTLR is a language recognition toolset which uses LL(*) grammer for parsing. There are language grammers available including C [4], JavaScript [5], Java [6] etc. We used antlr to create a parse tree for the C soruce 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 antler 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

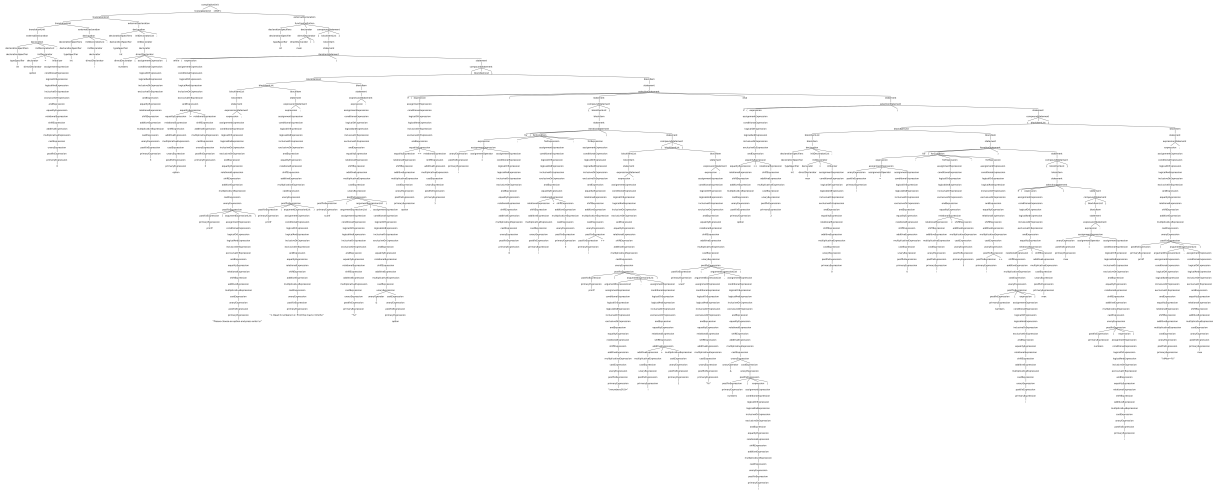

Figure 3.2: ANTLR parse tree created from C source at 2.1

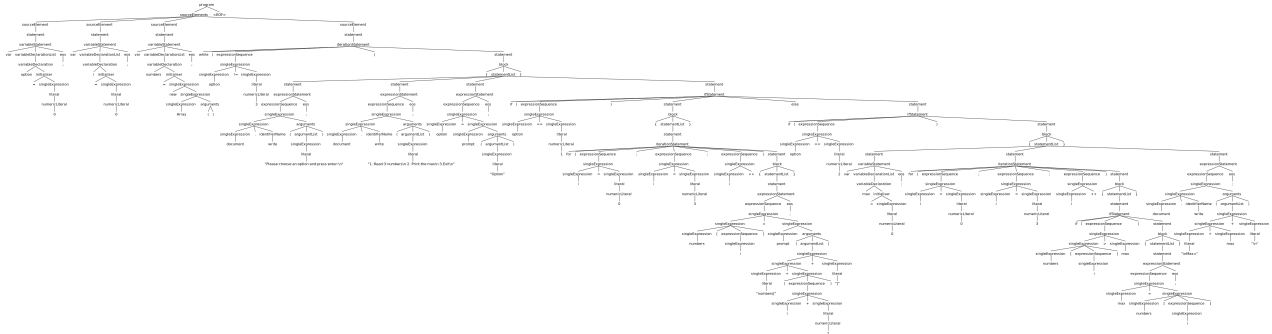

Figure 3.3: ANTLR parse tree created from JS source at 2.2

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.
[7]

[7] 3.2 and 3.3 has not been fully implemented due to time shortage.

4

Appendix

4.1 Owl soruce code for C source code defined in 2.1

```
<?xml version="1.0"?>
<rdf:RDF xmlns="https://semantic-web.netlify.com/
    ↪ c_source_ontology/"
     xml:base="https://semantic-web.netlify.com/
    ↪ c_source_ontology/"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xkos="http://rdf-vocabulary.ddialliance.org/xkos#"
     xmlns:dcterms="http://purl.org/dc/terms/"
     xmlns:c_source_ontology="https://semantic-web.netlify.com/
    ↪ c_source_ontology/">
    <owl:Ontology rdf:about="https://semantic-web.netlify.com/
    ↪ c_source_ontology/">
        <rdfs:comment xml:lang="en">An ontology that represents
    ↪  structural and imperative programming languages</
    ↪ rdfs:comment>
        <rdfs:label xml:lang="en">SoruceCodeOntology</
    ↪ rdfs:label>
    </owl:Ontology>



    <!--

    ↪ ///////////////////////////////////////////////////////////////////////////////
    ↪
    //
    // Object Properties
    //

```

```
    ↪ ////////////////////////////////////////////////////////////////////////////////
    ↪
25    -->
26
27
28
29
30    <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ conditions -->
31
32    <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/c_source_ontology/conditions">
33        <owl:inverseOf rdf:resource="https://semantic-web.
    ↪ netlify.com/c_source_ontology/has_condition"/>
34        <rdfs:domain>
35            <owl:Restriction>
36                <owl:onProperty rdf:resource="https://semantic-
    ↪ web.netlify.com/c_source_ontology/conditions"/>
37                <owl:someValuesFrom rdf:resource="https://
    ↪ semantic-web.netlify.com/c_source_ontology/
    ↪ ConditionalStructure"/>
38            </owl:Restriction>
39        </rdfs:domain>
40        <rdfs:range>
41            <owl:Restriction>
42                <owl:onProperty rdf:resource="https://semantic-
    ↪ web.netlify.com/c_source_ontology/conditions"/>
43                <owl:someValuesFrom rdf:resource="https://
    ↪ semantic-web.netlify.com/c_source_ontology/
    ↪ RepetitiveStructure"/>
44            </owl:Restriction>
45        </rdfs:range>
46    </owl:ObjectProperty>
47
48
49
50    <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ has_condition -->
51
52    <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/c_source_ontology/has_condition"/>
53
54
55
56    <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ has_type -->
57
58    <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/c_source_ontology/has_type">
59        <owl:inverseOf rdf:resource="https://semantic-web.
    ↪ netlify.com/c_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/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/
    ↪ dimension -->
109 
110     <owl:DatatypeProperty rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/dimension">
111         <rdfs:subPropertyOf rdf:resource="http://www.w3.org
    ↪ /2002/07/owl#topDataProperty"/>
112     </owl:DatatypeProperty>
113 
114 
115 
116     <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ name -->
117 
118     <owl:DatatypeProperty rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/name">
119         <rdfs:domain>
120             <owl:Class>
121                 <owl:unionOf rdf:parseType="Collection">
122                     <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/c_source_ontology/Constants"/>
123                     <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/c_source_ontology/DataType"/>
124                 </owl:unionOf>
125             </owl:Class>
126         </rdfs:domain>
127         <rdfs:range rdf:resource="http://www.w3.org/2001/
    ↪ XMLSchema#string"/>
128     </owl:DatatypeProperty>
129 
130 
131 
132     <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ source_code -->
133 
134     <owl:DatatypeProperty rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/source_code">
135         <rdfs:domain>
136             <owl:Class>
137                 <owl:unionOf rdf:parseType="Collection">
138                     <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/c_source_ontology/
    ↪ 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">
```

```
        <rdfs:subClassOf rdf:resource="https://semantic-web.
    ↪ netlify.com/c_source_ontology/ProgrammingStructure"/>
    </owl:Class>



    <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ SystemFunction -->

    <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↪ c_source_ontology/SystemFunction"/>



    <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ Variable -->

    <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↪ c_source_ontology/Variable"/>



    <!--

    ↪ ///////////////////////////////////////////////////////////////////////////////
    ↪
    //
    // Individuals
    //

    ↪ ///////////////////////////////////////////////////////////////////////////////
    ↪
      -->



    <!-- https://semantic-web.netlify.com/c_source_ontology/For
    ↪  -->

    <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/For">
        <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/RepetitiveStructure"/>
        <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/c_source_ontology/if"/>
        <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">for (i=0; i&lt;3; i++) {
        printf(&quot;\nnumbers[%i]=&quot;,i+1);scanf(&quot;%i&
    ↪ quot;,&amp;numbers[i]);
}</source_code>
    </owl:NamedIndividual>


```

```
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&lt;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/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(&amp;quot;Please choose an option and press enter:\n&
    ↪ amp;quot;);
296     printf(&amp;quot;1. Read 3 numbers\n 2. Print the max\n 3.Exit\
    ↪ n&amp;quot;);
297     scanf(&amp;quot;%i&amp;quot;,&amp;amp;option);
298     if (option==1) {
299         for (i=0; i&amp;lt;3; i++) {
300             printf(&amp;quot;\nnumbers[%i]=&amp;quot;,i+1);scanf(&amp;quot
    ↪ ;%i&amp;quot;,&amp;amp;numbers[i]);
301         }
302     } else if (option==2) {
303         int max = 0;
304         for (i=0; i&amp;lt;3; i++) {
305             if(numbers[i] &amp;gt; max) {
306                 max = numbers[i];
307             }
308         }
309         printf(&amp;quot;\nMax=%i&amp;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&amp;lt;3; i++) {
325             if(numbers[i] &amp;gt; max) {
326                 max = numbers[i];
327             }
328         }
329         printf(&amp;quot;\nMax=%i&amp;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&lt;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"/>
```

```
        <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">if(numbers[i] &gt; max) {
    max = numbers[i];
}</source_code>
    </owl:NamedIndividual>



    <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ int_array -->

    <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/int_array">
        <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/Array"/>
        <dimension rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#int">1</dimension>
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">integer array</name>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">Integer array</rdfs:label>
    </owl:NamedIndividual>



    <!-- https://semantic-web.netlify.com/c_source_ontology/max
    ↪  -->

    <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/max">
        <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/Variable"/>
        <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/c_source_ontology/else_if"/>
    </owl:NamedIndividual>



    <!-- https://semantic-web.netlify.com/c_source_ontology/
    ↪ numbers -->

    <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/c_source_ontology/numbers">
        <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/Variable"/>
        <has_type rdf:resource="https://semantic-web.netlify.
    ↪ com/c_source_ontology/int_array"/>
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">numbers</name>
        <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">int numbers[3]</source_code>
    </owl:NamedIndividual>


```

```
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(&amp;quot;Please choose an option
    ↪ and press enter:\n&amp;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(&amp;quot;1. Read 3 numbers\n 2.
    ↪ Print the max\n 3.Exit\n&amp;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(&amp;quot;\nnumbers[%i]=&amp;quot;,i+1);
    ↪ scanf(&amp;quot;%i&amp;quot;,&amp;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(&amp;quot;\nMax=%i&amp;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(&amp;quot;%i&amp;quot;,&amp;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 soruce 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/
    ↪ conditions -->
31
32      <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/js_source_ontology/conditions">
33          <owl:inverseOf rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/has_condition"/>
34          <rdfs:domain>
35              <owl:Restriction>
36                  <owl:onProperty rdf:resource="https://semantic-
    ↪ web.netlify.com/js_source_ontology/conditions"/>
37                  <owl:someValuesFrom rdf:resource="https://
    ↪ semantic-web.netlify.com/js_source_ontology/
    ↪ ConditionalStructure"/>
38              </owl:Restriction>
39          </rdfs:domain>
40          <rdfs:range>
41              <owl:Restriction>
42                  <owl:onProperty rdf:resource="https://semantic-
    ↪ web.netlify.com/js_source_ontology/conditions"/>
43                  <owl:someValuesFrom rdf:resource="https://
    ↪ semantic-web.netlify.com/js_source_ontology/
    ↪ RepetitiveStructure"/>
44              </owl:Restriction>
45          </rdfs:range>
46      </owl:ObjectProperty>
47
48
49
50      <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ has_condition -->
51
52      <owl:ObjectProperty rdf:about="https://semantic-web.netlify
    ↪ .com/js_source_ontology/has_condition"/>
53
54
55
56      <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ 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/
    ↪ dimension -->
109
110  <owl:DatatypeProperty rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/dimension">
111      <rdfs:subPropertyOf rdf:resource="http://www.w3.org
    ↪ /2002/07/owl#topDataProperty"/>
112  </owl:DatatypeProperty>
113
114
115
116  <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ name -->
117
118  <owl:DatatypeProperty rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/name">
119      <rdfs:domain>
120          <owl:Class>
121              <owl:unionOf rdf:parseType="Collection">
122                  <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/Constants"/>
123                  <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/DataType"/>
124              </owl:unionOf>
125          </owl:Class>
126      </rdfs:domain>
127      <rdfs:range rdf:resource="http://www.w3.org/2001/
    ↪ XMLSchema#string"/>
128  </owl:DatatypeProperty>
129
130
131
132  <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ source_code -->
133
134  <owl:DatatypeProperty rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/source_code">
135      <rdfs:domain>
136          <owl:Class>
137              <owl:unionOf rdf:parseType="Collection">
138                  <rdf:Description rdf:about="https://
```

```
    ↪ semantic-web.netlify.com/js_source_ontology/
    ↪ ConditionalStructure"/>
139                 <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/Constants"/>
140                 <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/DataType"/>
141                 <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/Operator"/>
142                 <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/
    ↪ ProgrammingStructure"/>
143                 <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/
    ↪ RepetitiveStructure"/>
144                 <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_source_ontology/
    ↪ SystemFunction"/>
145                 <rdf:Description rdf:about="https://
    ↪ semantic-web.netlify.com/js_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/js_source_ontology/
    ↪ Array -->
167
168   <owl:Class rdf:about="https://semantic-web.netlify.com/
    ↪ 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&lt;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&lt;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&lt;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&lt;3; i++) {
295             if(numbers[i] &gt; 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.netlify.
    ↪ 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.netlify.
    ↪ 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/
    ↪ XMLSchema#string">else if (option==2) {
324         var max = 0;
325         for (i=0; i&lt;3; i++) {
326             if(numbers[i] &gt; max) {
327                 max = numbers[i];
```

```
328             }
329         }
330         document.write(&amp;quot;\nMax=%i&amp;quot;,max);
331     }</source_code>
332     </owl:NamedIndividual>
333 
334 
335 
336     <!-- https://semantic-web.netlify.com/js_source_ontology/i
    ↪ -->
337 
338     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/i">
339         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/Variable"/>
340         <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/While"/>
341         <is_type_of rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/var"/>
342         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">i</name>
343         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">int i</source_code>
344         <rdfs:label rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">i</rdfs:label>
345     </owl:NamedIndividual>
346 
347 
348 
349     <!-- https://semantic-web.netlify.com/js_source_ontology/if
    ↪  -->
350 
351     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/if">
352         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/ConditionalStructure"/>
353         <is_else_branch_of rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/else_if"/>
354         <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/While"/>
355         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">if (option==1) {
356         for (i=0; i&amp;lt;3; i++) {
357             document.write(&amp;quot;\nnumbers[%i]=&amp;quot;,i+1);
    ↪ scanf(&amp;quot;%i&amp;quot;,&amp;amp;numbers[i]);
358         }
359     }</source_code>
360         <rdfs:label rdf:datatype="http://www.w3.org/2001/
    ↪ 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/
    ↪ option -->
399
400     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/option">
401         <rdf:type rdf:resource="https://semantic-web.netlify.
```

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

```
436         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/SystemFunction"/>
437         <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/While"/>
438         <is_type_of rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/Void"/>
439         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">document.write</name>
440         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">document.write(&quot;Please choose an
    ↪ option and press enter:\n&quot;);</source_code>
441     </owl:NamedIndividual>
442
443
444
445     <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ write2 -->
446
447     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/write2">
448         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/SystemFunction"/>
449         <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/While"/>
450         <is_type_of rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/Void"/>
451         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">document.write</name>
452         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">document.write(&quot;1. Read 3 numbers\
    ↪ n 2. Print the max\n 3.Exit\n&quot;);</source_code>
453     </owl:NamedIndividual>
454
455
456
457     <!-- https://semantic-web.netlify.com/js_source_ontology/
    ↪ write3 -->
458
459     <owl:NamedIndividual rdf:about="https://semantic-web.
    ↪ netlify.com/js_source_ontology/write3">
460         <rdf:type rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/SystemFunction"/>
461         <is_included_in rdf:resource="https://semantic-web.
    ↪ netlify.com/js_source_ontology/else_if"/>
462         <is_type_of rdf:resource="https://semantic-web.netlify.
    ↪ com/js_source_ontology/Void"/>
463         <name rdf:datatype="http://www.w3.org/2001/XMLSchema#
    ↪ string">document.write</name>
464         <source_code rdf:datatype="http://www.w3.org/2001/
    ↪ XMLSchema#string">document.write(&quot;\nMax= &quot; +
    ↪ max + &quot;\n&quot;);</source_code>
465     </owl:NamedIndividual>
466 </rdf:RDF>
467
```

```
468
469
470 <!-- Generated by the OWL API (version 4.5.9.2019-02-01
    ↪ T07:24:44Z) https://github.com/owlcs/owlapi -->
```

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