Homework 1: Syntax

CSC 600-01 Programming Languages

Spring 2017

Emilio Quiambao

2 / 12 / 2017

1. Syntax Definitions

(a) Natural number

<natural number> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | <natural number> <digit>

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

(b) Unsigned integer

<unsigned int> ::= <digit> | <unsigned int> <digit>

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

(c) Integer

<int> ::= 0 | <sign> <number>

<sign> ::= - | <empty>

<empty> ::=

<number> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | <number> <digit>

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

(d) Odd number

<odd number> ::= <sign> <number> <odd digit> | <sign> <odd digit>

<sign> ::= <empty> | -

<empty> ::=

<number> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | <number> <digit>

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<odd digit> ::= 1 | 3 | 5 | 7 | 9

(e) Even number

<even number>	::=	<sign> <number> <even digit> |
		<sign> <even digit> | 0
<sign>	::=	<empty> | -
<empty>	::=	
<number>	::=	1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | <number> <digit>
<digit>	::=	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<even digit>	::=	2 | 4 | 6 | 8

(f) Integer divisible by five

<int divisible by five>	::=	<sign> <number> <div tail> | 0 | <sign> 5
<sign>	::=	<empty> | -
<empty>	::=	
<number>	::=	1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | <number> <digit>
<digit>	::=	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<div tail>	::=	0 | 5

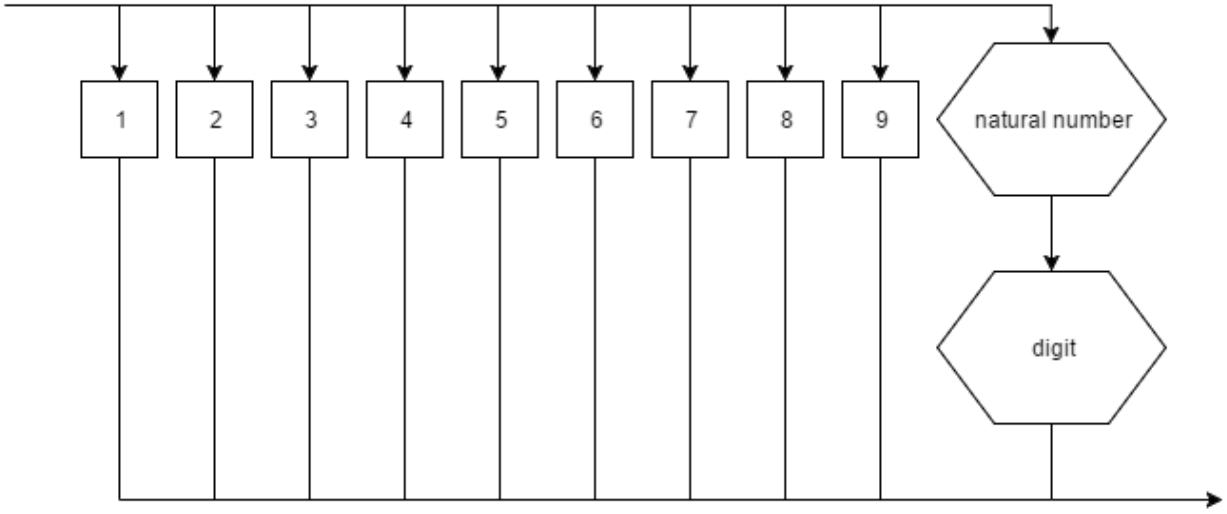
2. Syntax Diagrams

The following diagrams flow top to bottom, left to right.
They cannot go backwards unless pointed otherwise

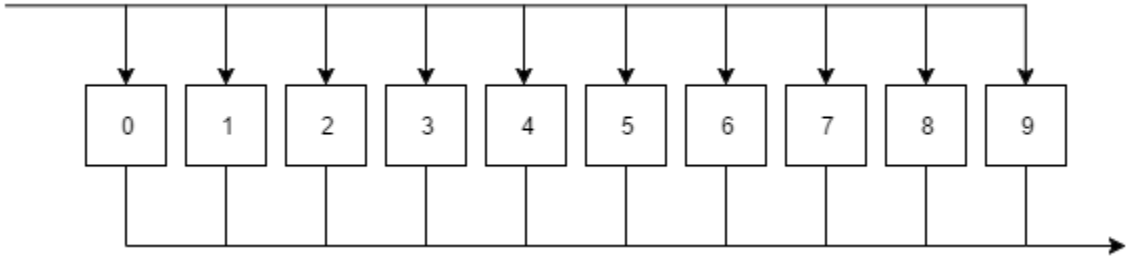
Boxes represent tokens / terminal symbols.
Hexagons represent objects / metalinguistic variables.

(a) Natural number

Natural Number Diagram

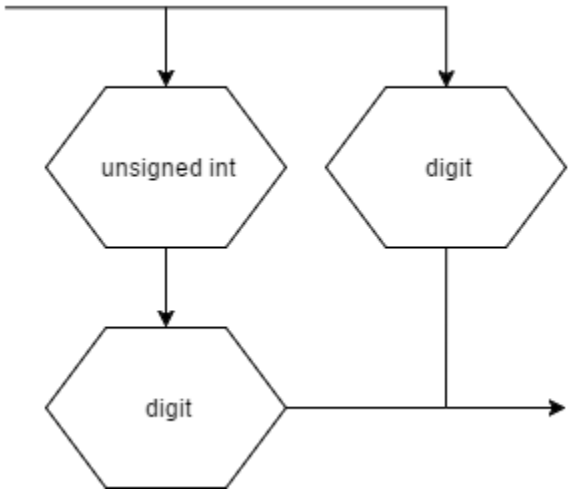


Digit Diagram

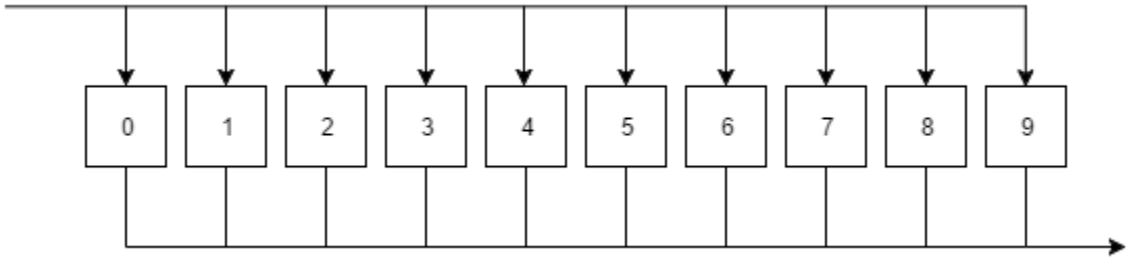


(b) Unsigned integer

Unsigned Integer Diagram

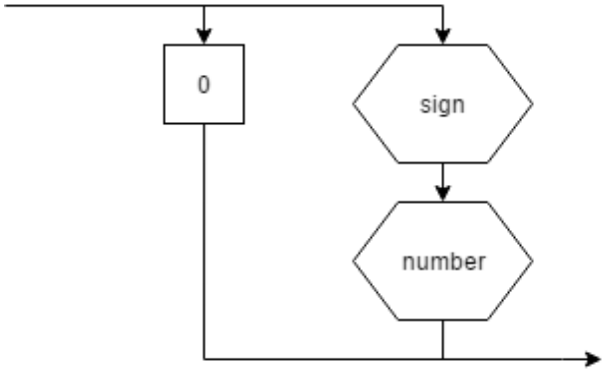


Digit Diagram

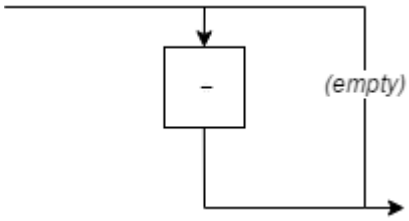


(c) Integer

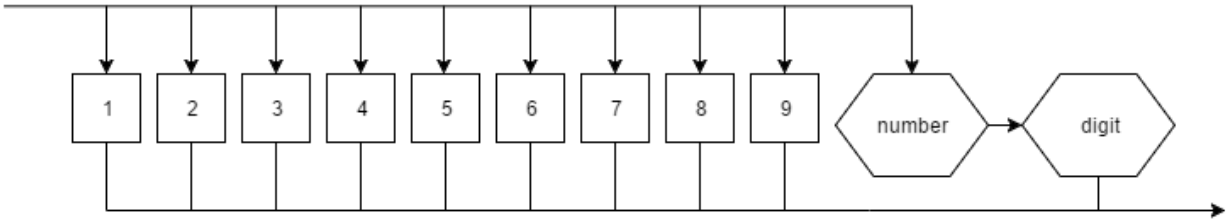
Integer Diagram



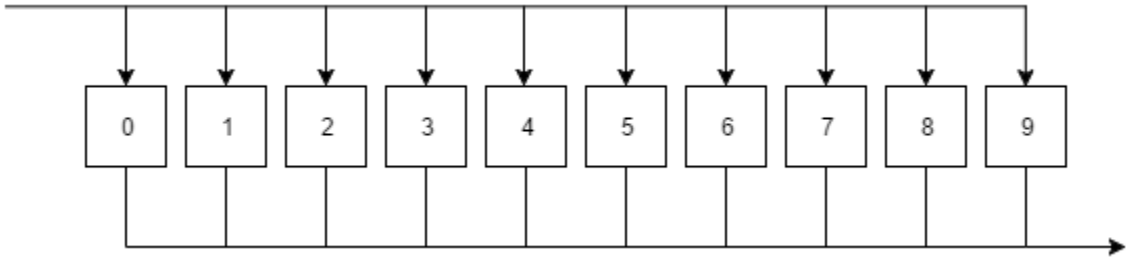
Sign Diagram



Number Diagram

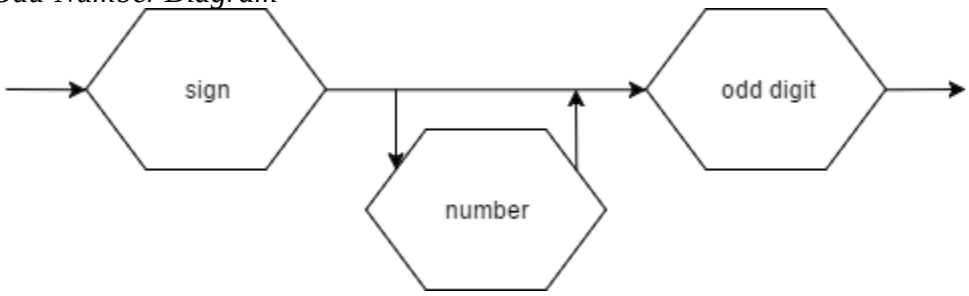


Digit Diagram

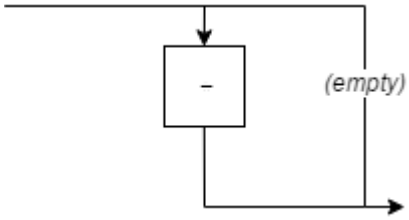


(d) Odd number

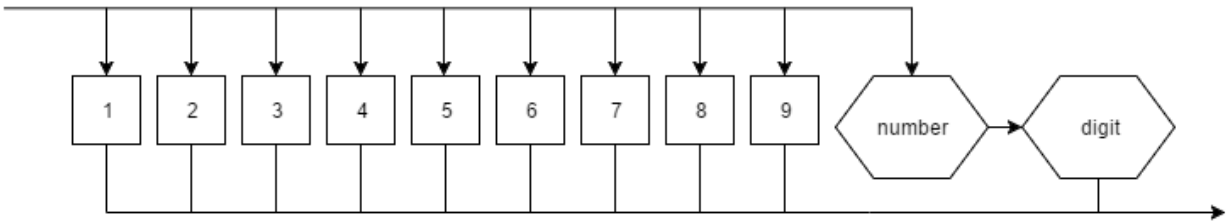
Odd Number Diagram



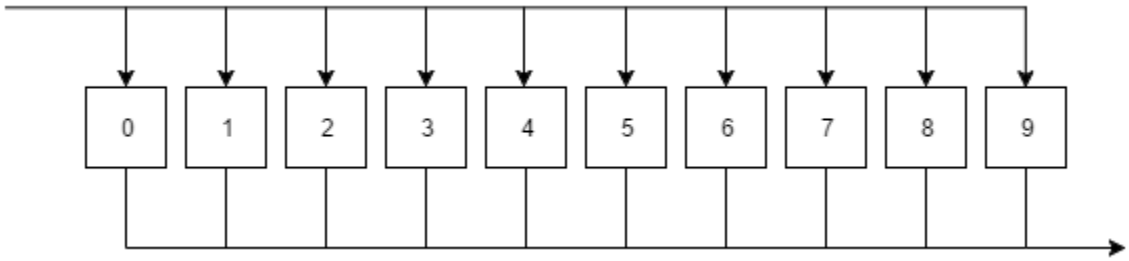
Sign Diagram



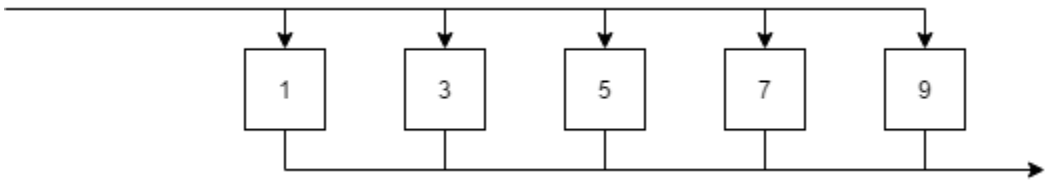
Number Diagram



Digit Diagram

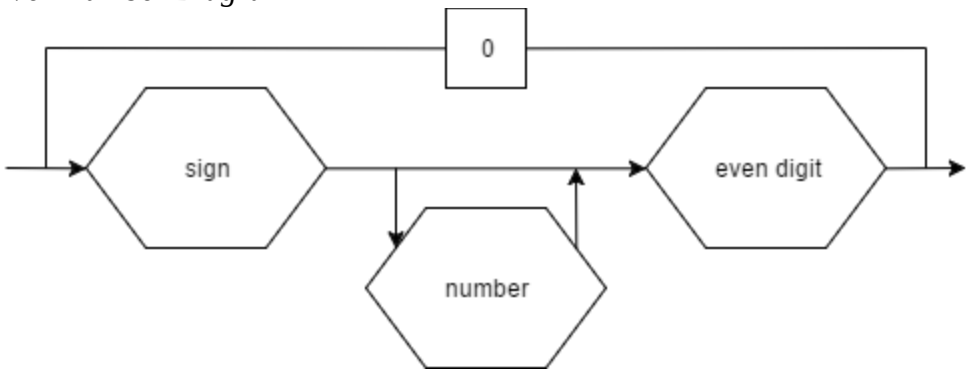


Odd Digit Diagram

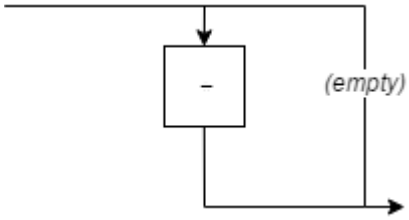


(e) Even number

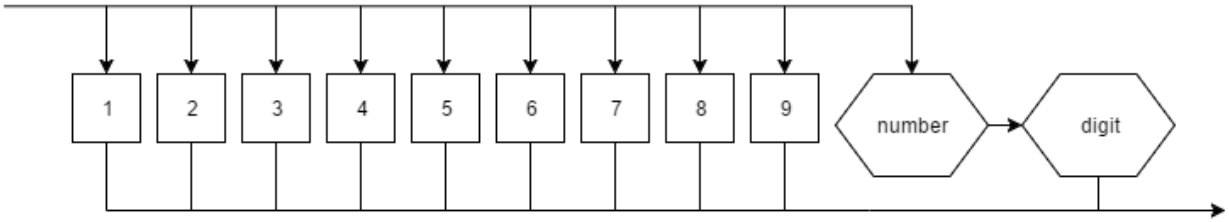
Even Number Diagram



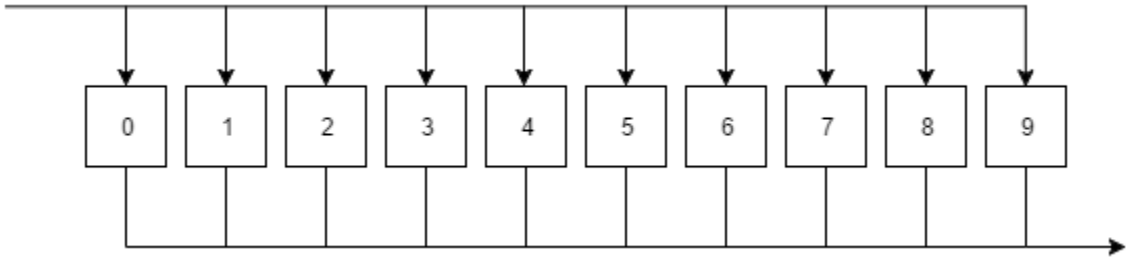
Sign Diagram



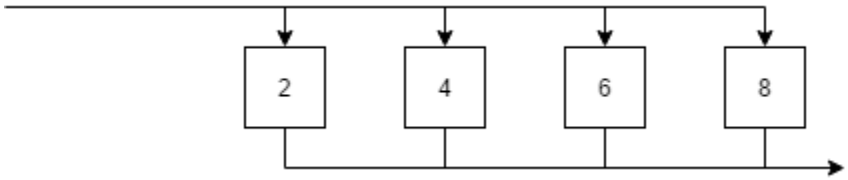
Number Diagram



Digit Diagram

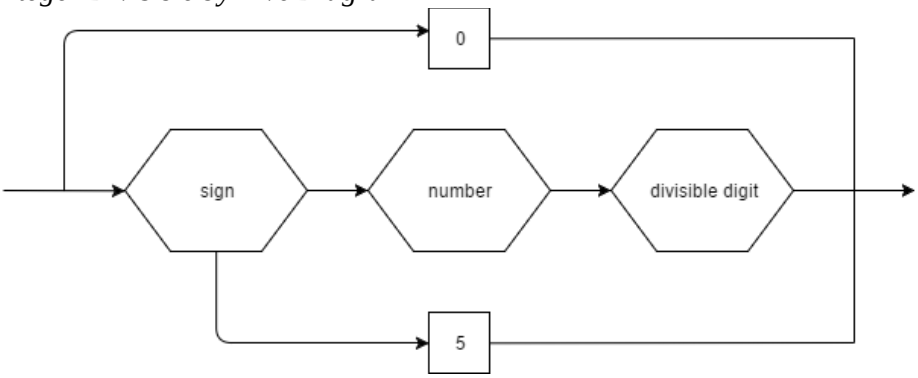


Even Digit Diagram

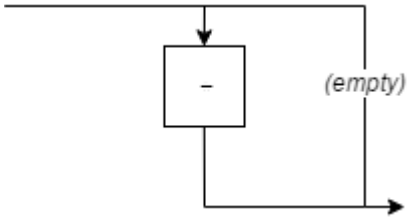


(f) Integer divisible by five

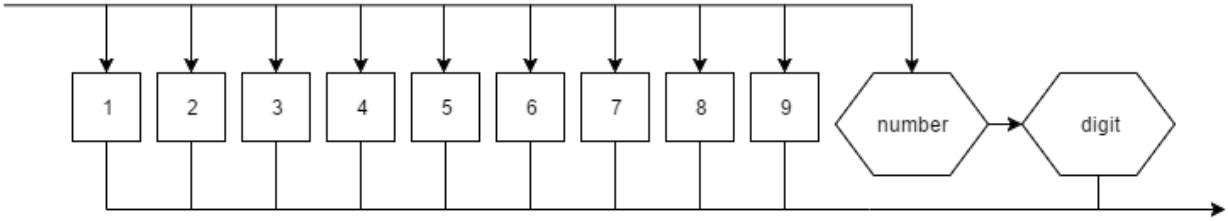
Integer Divisible by Five Diagram



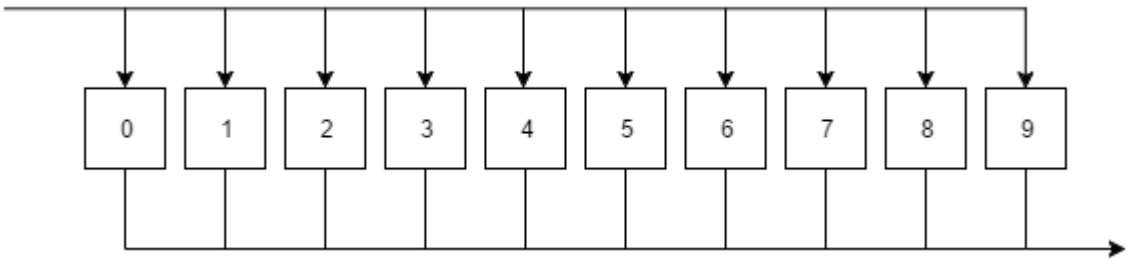
Sign Diagram



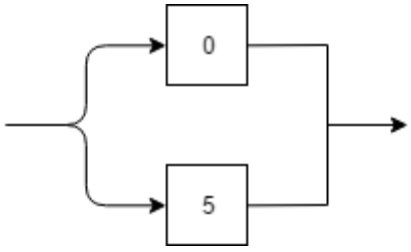
Number Diagram



Digit Diagram



Divisible Digit Diagram



3. Syntax of Input Statement in C++

<input statement> ::= cin <input> <semicolon>

<input> ::= <arithmetic right shift> <identifier> |
<arithmetic right shift> <expression> |
<input> <input>

<semicolon> ::= ;

<arithmetic right shift> ::= >>

<identifier> ::= <letter> | <identifier> <char>

<expression> ::= <operand> | <operand> <operation>
| [<expression>] | (<expression>)

<operand> ::= <number> | <identifier> | <function> | <array>

<operation> ::= + <operand> |
- <operand> |
* <operand> |
/ <operand> |
% <operand> |
+ <operand> <operation> |
- <operand> <operation> |
* <operand> <operation> |
/ <operand> <operation> |
% <operand> <operation> |

<number> ::= <sign> <decimal> <sci notation>

<sign> ::= - | <empty>

<decimal> ::= <unsigned int> | <unsigned int> . <unsigned int>

<sci notation> ::= e <sign> <number> | <empty>

<empty> ::=

<unsigned int> ::= <digit> | <unsigned int> <digit>

<char> ::= <letter> | <digit>

<letter>	::=	a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | R | S | T | U | V | W | X | Y | Z | _
<digit>	::=	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<array>	::=	<identifier> <index>
<index>	::=	[<expression>] | <index> [<expression>]
<function>	::=	<identifier> (<argument>)
<argument>	::=	<expression> | <argument>, <expression>

4. Syntax of Input Statement in C++

<input statement>	::=	cout <output> <semicolon>
<output>	::=	<arithmetic left shift> <identifier> | <arithmetic left shift> <expression> | <arithmetic left shift> "<string>" | <arithmetic left shift> '<character>' | <arithmetic left shift> <array> | <arithmetic left shift> <function> | <arithmetic left shift> <relation expr>| <arithmetic left shift> <logical expr> | <output> <output>
<arithmetic left shift>	::=	<<
<string>	::=	<character> | <string> <character> | <expression> + <string> | <string> + <string>
<character>	::=	<letter> | <digit> | <whitespace> | ! | @ | ` | ~ | # | … + | { | } | [|] | ? | < | >… (etc.)
<whitespace>	::=	" "

<relation expr>	::=	<rel operand> <rel operator> <rel operand> |
		<rel operand> <rel operator> <relation expr>
<rel operand>	::=	<identifier> | <expression> | <array> | <function>
<rel operator>	::=	< | > | != | <= | >= | ==
<logical expr>	::=	<negation> <logic operand> |
		<logic operand> <logic operator> <logic operand> |
		<logic operand> <logic operator> <logical expr>
<negation>	::=	!
<logic operand>	::=	true | false | <identifier> | <expression> | <function> | <array>
<logic operator>	::=	&& | ||