**Basic data type:**

1. Number (Double, Integer)
2. Table (1-D canonical array)
   1. Fixed size based on declaration: a={2, 3, 4, 5} can have at most that size

Variables:

1. Global variables (not inside functions)
   1. Single and multiple definition and initialization
2. Local variables (only in functions and flow control instructions and local scopes)

2.1. Single and multiple definition

2.2. Single and multiple definition and inline initialization

3. Reassignment of global and local variables (within same type)

**Operators:**

1. Arithmetic
   1. Sum +
   2. Subtraction –
   3. Multiplication \*
   4. Dision /
   5. Exponentiation ^ (of immediates through ^ operatore, with math.pow(x, y) for anything)
2. Logical
   1. AND and
   2. OR or
   3. NOT not (not implemented)
3. Relational
   1. Equality
   2. Inequality
   3. Less than
   4. Greater than
   5. Less than or equal
   6. Greater than or equal

**Flow Control Instructions:**

1. For loop
2. While loop
3. Repeat-until
4. If then else
5. Nested Flow Control Instruction (any depth and all)

Loop Condition Supported: Single numbers, single variables, Boolean expressions (of any length), single array elements, mathematical expressions.

**Functions:**

1. Function declaration (anywhere in the code) (only NUMBER as parameter and return value must be a NUMBER)
2. Print function (Only string or only multiple numbers/variables)
3. Print(string.format()) (c-like printf) Inclusion of external modules though require keywork (function must be called with name.function)
4. “require” function implemented. It is possible to import file (libraries) and used them through library.namefunction(). Support also for global variables declared in libraries and error in case of multiple declarations.

**Semantic error supported:**

1. Variable not declared
2. Operation not supported by compiler(NOT)
3. Redeclaration of a variable into an array (limitation of compiler)
4. Array access to array not declared
5. Array access to variable (not array)
6. Redeclaration of function
7. Invocation of function not declared
8. Wrong number of parameters to function

**Syntactical warning supported:**

1. Error in if condition
2. Error in assignment of a variable
3. Missing } in array declaration
4. Missing ) in function declaration
5. Missing return value of a function

**Syntactical error supported**

1. Expression between () not correct
2. Wrong variables list
3. Wrong loop variable initialization
4. Assign an array to a variable and vice versa
5. Declaration of global variables inside functions (Compiler limitation)
6. Print function without parameters
7. Pass an array to a function (Compiler limitation)
8. Duplication of require of a file