**Basic data type:**

1. Number (Double, Float)
2. table (1-D canonical array)
   1. Fixed size based on declaration: a={2, 3, 4, 5} can have at most that size
   2. Array elements can be used to initialize variable (also multiple initiliazations) a, b=c[1], c[0]
   3. Single elements can be initialized

Variables:

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

2.1. Single and multiple definition

2.2. Single and multiple definition and inline initialiation

3. Reassignment of global and local variables

**Operators:**

1. Arithmetic
   1. Sum +
   2. Subtraction –
   3. Multiplication \*
   4. Dision /
   5. Exponentiation ^
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)

Functions

1. Function declaration (anywhere in the code) (only NUMBER as parameter and return value must be a number)
2. Print function (Only string
3. Definition of what to implement
4. Do parser
5. Translate in LLVM

Error recognition:

During variable declaration, I noticed that as variables have no type, they can store whatever they want. So to solve the problem, one could redeclare a global variable each time the type changes, solving the problem.