

In this project, I created my programming language according to given rules, and I added extra basic concepts or features. I do not use function, union and struct definition. The basics of my programming language belongs to c language but we have a different type of rules like start with "BRAIN" , and end the program with "DEAD" letter. Consequently, we have a lot of features, I am going to explain all of them below.

BNF

Arithmetic Operators:

<arithmetic operators> ::= + | - | / | *

Comparison Operators:

<comparison operators> ::= < | > | <= | >= | !=

<digits> ::= <digit> | <digits><digit>

<digit> ::= 0 | <non zero digit>

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

<floating> ::= <digits>.<digits>

<integer> ::= <sign>? <digits>

<number> ::= <integer> | <floating>

<sign> ::= + | -

<letter> ::=
 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|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|

<chars> ::= <digits> | <letters>

<special chars> ::= \n

<punctuation> ::= . | : | ; | () | [] | { }

<text> ::= <text> | <chars> | <punctuation>

<string> ::= "<text>"

<expression> ::= <variable name> | <number>

<any statement> ::= <if statement> | <assignment
 statement> | <loop statement> | <arithmetic operators>

Variable Names:

<variable name> ::= <variable name> | <letters><chars> |
 <letters>

Assignment Statements:

<declaration statement> ::= <variable type><variable name> ;
 | <variable type><assignment statement> ;

<assignment statement> ::= <left hand side> <assignment
 operator><assignment>

<left hand side> ::= <variable name>

<assignment operator> ::= = | += | -=

Loops:

<loop statement> ::= <while loop> | <for loop>

<while loop> ::= while (<conditional expression>) struct

<for loop> ::= for(<assignment statement>; conditional expression;<assignment statement>) struct

<struct> ::= {stmts} | { }

Conditional Statements:

<if statement> ::= <matched> | <unmatched>

<matched> ::= if(<boolean expression>) body

<unmatched> ::= if(<conditional expression>) then <if statement> | if(<conditional expression>) then <matched> else <unmatched>

RULES

Program must write "BRAIN{" and end with "}DEAD" words. BRAIN refers to the meaning of the classic word begin, and DEAD refers to meaning of the classic word end, also the user must place the program between the left curly bracket after BRAIN and right curly bracket before DEAD.

There is four data type string(str), float, integer(int), and const type to be used for variable names.

All data types (basic types) must declare between square brackets.

After every assignment statement, variable definition, print statement users must use "^" symbol to switch to new line.

int data type can be signed or unsigned (+, -).

Punctuation cannot be included in identifier data type.

Identifiers cannot be only digits, and cannot start with digit.

Conditional expression are used for comparing if two variables are equal, not equal, less or more than each other, or not.

Definition of Language Constructs:

For "if statement", conditional expression part is written between left and right brackets. After left curly bracket users can enter the condition and finish with right curly bracket.

For "for loop", all the statement are written left and right brackets and between all the statements, there must be a semicolon. After left curly bracket users can enter the condition and finish with right curly bracket.

For "while loop", conditional expression part is written between left and right brackets and the statement should be given after left curly bracket and finish with right curly bracket.

For "print statement" expression part is written between left curly bracket and right curly bracket.

For "assignment statement" colon and equal operator should

be placed between left and right expressions. ($i ::= 1$)