**MINILANGUAGE SPECIFICATION**

1. **LANGUAGE DEFINITION**

* Alphabet:

<letter> ::= "A" | "B" | ... | "Z" | "a" | "b" | ... | "z" ;

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

<symbol> ::= “\_” ;

* Lexicon:
* Special Symbols:

<operator> ::= "+" | "-" | "\*" | "/" | ":=" | "<" | "<=" | "=" | ">=" ;

<separator> ::= "[" | "]" | "{" | "}" | ":" | ";" | " " ;

* Keywords:

<keyword> ::= "int" | "float" | "var" | "if" | "else" | "while" | "read" | "write" ;

* Identifiers:

<identifier> ::= <letter> , { <letter> | <digit> } ;

* Constants:

<integer> ::= [ "+" | "-" ] , <digit> , { <digit> } ;

<float> ::= [ "+" | "-" ] , <digit> , { <digit> } , "." , <digit> , { <digit> } ;

<string> ::= '"' , { <char> } , '"' ;

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

1. **SYNTAX**

* Program Structure:

<program> ::= "VAR" , <decllist> , ";" , <cmpdstmt> , "." ;

<decllist> ::= <declaration> | <declaration> , ";" , <decllist> ;

<declaration> ::= <IDENTIFIER> , ":" , <type> ;

<type1> ::= "BOOLEAN" | "CHAR" | "INTEGER" | "REAL" ;

<arraydecl> ::= "ARRAY" , "[" , <nr> , "]" , "OF" , <type1> ;

<type> ::= <type1> | <arraydecl> ;

<cmpdstmt> ::= "BEGIN" , <stmtlist> , "END" ;

<stmtlist> ::= <stmt> | <stmt> , ";" , <stmtlist> ;

<stmt> ::= <simplstmt> | <structstmt> ;

<simplstmt> ::= <assignstmt> | <iostmt> ;

<assignstmt> ::= <IDENTIFIER> , ":=" , <expression> ;

<expression> ::= <expression> , "+" , <term> | <term> ;

<term> ::= <term> , "\*" , <factor> | <factor> ;

<factor> ::= "(" , <expression> , ")" | <IDENTIFIER> ;

<iostmt> ::= "READ" , "(" , <IDENTIFIER> , ")" | "WRITE" , "(" , <IDENTIFIER> , ")" ;

<structstmt> ::= <cmpdstmt> | <ifstmt> | <whilestmt> ;

<ifstmt> ::= "IF" , <condition> , "THEN" , <stmt> , [ "ELSE" , <stmt> ] ;

<whilestmt> ::= "WHILE" , <condition> , "DO" , <stmt> ;

<condition> ::= <expression> , <RELATION> , <expression> ;

<RELATION> ::= "<" | "<=" | "=" | "<>" | ">=" | ">" ;

1. **b) Write a small program into your programming language (for example: find the first k prime numbers, print all prime numbers less than k)**

VAR

k : int;

count : int;

num : int;

i : int;

is\_prime : int;

BEGIN

READ(k);

count := 0;

num := 2;

WHILE count < k DO

is\_prime := 1;

i := 2;

WHILE i \* i <= num DO

IF num % i = 0 THEN

is\_prime := 0;

END;

i := i + 1;

END;

IF is\_prime = 1 THEN

WRITE(num);

count := count + 1;

END;

num := num + 1;

END;

END.