

## Language Definition

### 1. Alphabet

2.

1. Upper (A-Z) and lower (a-z) case letters of the English alphabet

2. Decimal digits (0-9)

### 2. Lexicon

1. Special symbols, representing:

1. Operators: + - \* / ^ = < > <= >=

2. Separators: space # ; { } ( ) [ ]

3. Reserved words: char int while else if read write

2. Identifiers (a sequence of letters & digits, such that the first character is a letter; the rule is:

1. identifier = letter {letter | digit}

2. letter = "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"

3. digit = "0" | "1" | ... | "9"

### 3. Constants

1. Integer

int = sign non-zero-digit | digit | sign non-zero-digit sequence-of-digits | non-zero-digit sequence-of-digits

sign = "+" | "-"

non-zero-digit = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

2. Character

character := 'letter' | 'digit'

### 3. Syntax

#### A. Syntactical Rules

■ program = "main" "(" "(" ")" "{" compseq "}"

■ list = "[" [ factor {"," factor} ] "]"

■ stmt = simplestmt | structstmt

- `simplestmt = assignstmt | iostmt | condstmt`
- `compseq = stmt {compseq}`
- `assignstmt = IDENTIFIER "=" expression`
- `expression = factor | factor OPERATOR expression`
- `factor = IDENTIFIER | CONSTANT`
- `iostmt = "read" IDENTIFIER | "write" IDENTIFIER | "write" CONSTANT`
- `condstmt = cmpdstmt | ifstmt | whilestmt`
- `ifstmt = "if" "(" ("condition") "{" stmtseq "}" ["else" "{" stmt "}]`
- `whilestmt = "while" "(" ("condition") "{" stmtseq "}"`
- `condition = expression RELATION expression`

## B. Lexical Rules

- `identifier = letter {letter | digit}`
- `letter = "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"`
- `digit = "0" | "1" | ... | "9"`
- `RELATION = "<" | "<=" | "==" | ">=" | ">"`
- `OPERATOR = "+" | "-" | "*" | "/" | "%" | ...`

The tokens are codified according to the following table:

- identifiers - code 0
- constants - code 1
- reserved words: each word has its own code
- operators: each operator has its own code
- separators: each separator has its own code

Codification:

Token type	code
identifier	0
constant	1
char	2

int	3	
-----		
while	4	
-----		
else	5	
-----		
if	6	
-----		
main	7	
-----		
read	8	
-----		
write	9	
-----		
+	10	
-----		
-	11	
-----		
*	12	
-----		
/	13	
-----		
%	14	
-----		
space	15	
-----		
#	16	
-----		
;	17	
-----		
{	18	
-----		
}	19	
-----		
(	20	
-----		
)	21	
-----		
[	22	
-----		
]	23	
-----		