Language Specification:

1. Language Definition:

1.1 Alphabet:

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

b. Decimal digits (0-9);

1.2 Lexic:

a. Special symbols, representing:

- operators:

arithmetic: +, -, \*, /, %

assignment: =

equality testing: ==, !=

logical: !, &&, ||

order relations: <, <=, >, >=

- separators [ ] { } ; ‘ ’ space

- reserved words:

char int if else printf scanf while main return

b. Identifiers

- a sequence of letters and digits, such that the first character is a letter; the rule is:

identifier = letter[{letter | digit}]

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

digit = non\_zero\_digit | zero\_digit

non\_zero\_digit = "1" | … | "9"

zero\_digit = "0"

sign = "-"

c. Constants

1. integer:

integer = zero\_digit | [sign] non\_zero\_digit {digit}

positive\_integer = non\_zero\_digit {digit}

2. character:

character = ‘letter’ | ‘digit’

2.2 Syntax:

The words - predefined tokens are specified between " and ":

a) Syntactical rules:

program = “int main() {“ statement\_list “ return 0;}”

statement\_list = statement {statement}

statement = declaration | simple\_statement | complicated\_statement

declaration = type identifier”;” | type identifier”[”positive\_integer”];”

type = “int” | “char”

simple\_statement = assignment | out\_statement | in\_statement

assignment = identifier “=” expression “;” | identifier”[”positive\_integer”]” “=” expression “;”

expression = [!] (term | expression operation expression | “(” expression operation expression “)”)

operation = “+” | “-” | “\*” | “/” | “%”

term = identifier | constant

out\_statement = "printf" "(" term ")" ";"

in\_statement = "scanf" "("identifier ")" ";"

complicated\_statement = if\_statement | while\_statement

if\_statement = “if” “(“ condition “)” “{”statement\_list“}” [“else{”statement\_list“}”]

condition = expression relation expression

while\_statement = “while(” condition “) {” statement\_list“}”

b) Lexical rules:

identifier = letter | letter{letter}{digit}

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

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

relation = "<" | "<=" | "==" | "!=" | ">=" | ">" | "&&" | "||"

|  |  |
| --- | --- |
| Token type | Code |
| identifier | 0 |
| constant | 1 |
| char | 2 |
| int | 3 |
| if | 4 |
| else | 5 |
| printf | 6 |
| scanf | 7 |
| while | 8 |
| main | 9 |
| return | 10 |
| + | 11 |
| - | 12 |
| \* | 13 |
| / | 14 |
| % | 15 |
| < | 16 |
| <= | 17 |
| = | 18 |
| >= | 19 |
| > | 20 |
| == | 21 |
| && | 22 |
| || | 23 |
| ! | 24 |
| != | 25 |
| [ | 26 |
| ] | 27 |
| { | 28 |
| } | 29 |
| ( | 30 |
| ) | 31 |
| ; | 32 |
| space | 33 |