

```
program = "start!" {declaration | compound_statement} "end!"
declaration = type identifier ";"
type = usertype | "integer" | "string" | "char" | "boolean" | "array"
compound_statement = statement {statement}
statement = (inputstmt | outputstmt | typedefstmt | assignstmt |
whilestmt | ifstmt | elsestmt) ";"
inputstmt = "input" identifier
outputstmt = "output" (identifier | constant)
typedefstmt = "typedef" type identifier
assignstmt = identifier "=" (constant | identifier)
ifstmt = "if" condition ":" compound_statement ["else:" elsestmt]
elsestmt = compound_statement
whilestmt = "while" condition "do:" compound_statement ";"
condition = (identifier | const) relations (identifier | const)
relations = "==" | "!=" | "<=" | ">=" | "<" | ">"
```

```
+  
-  
*  
/  
%  
<=  
>=  
==  
!=  
!  
<  
>  
[  
]  
(  
)  
/  
"  
,  
.  
\n  
array  
char  
const  
do  
else  
if  
int  
of  
output  
input  
for  
break  
then  
var  
string  
boolean  
true  
false  
while  
typedef
```

Alphabet

- a. Upper (A-Z) and lower case letters (a-z) of the English alphabet
- b. Underline character '_';
- c. Decimal digits (0-9);

Lexic:

- a. Special symbols, representing:
 - operators + - * / % = < > <= = >= != == !
 - separators [] { } : ; space () ,
 - reserved words:

array char const do else if int of input ouput for break then var
string boolean true false while typedef start! end! start end

b. identifiers

- a sequence of letters and digits, such that the first character is a letter;
- the number of characters is ≤ 256 ;
- the rule is:

identifier = letter {letter | digit | "_"}

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

digit_excluding_zero = "1" | "2" | "3" | "4" | "5" | "6" | "7" |
"8" | "9"

digit = "0" | digit_excluding_zero

c. constants

1. integer - rule:

integer = ["-"] natural_number | natural_number

natural_number = digit_excluding_zero {digit} | digit

2. character

character = 'letter' | 'digit' | 'symbol' | '_'

symbol = "\n" | " " | "." | "!" | "?" | "\"" | ":" | ";" | "|" |
"/" | "" | "=" | "+" | "(" | ")" | "&" | "*" | "-" | "_" | "=" | "+" |
"[" | "]"
| "{" | "}" | "<" | ">" | "\t"

3. string

string = "'constchar'"

```
constchar = char{constchar}
```

```
char = letter | digit | symbol | "_"
```

4.boolean

```
constbool = ("true" | "false")
```

5.array

```
constarray = "{"({integer} | {character} | {string} |  
{boolean})"{"
```