

letter ::= "a"|"b"|...|"z"|"A"|...|"Z"  
digit ::= "0"|"1"|"2"|...|"9"  
non\_zero\_digit ::= "1"|"2"|...|"9"

symbols ::= "\_"

unary\_operator ::= "!"  
binary\_operator ::= "+" | "-" | "\*" | "/" | "^" | "%" |  
| "and" | "or" |  
| ">" | "<" | ">=" | "<=" | "!=" | "==" |  
| "," // string concatenation

operator ::= "=" | unary\_operator | binary\_operator

separators ::= "[" | "]" | "{" | "}" | ";" | "space" | "newline"

identifier ::= letter{letter|digit|symbol}\{0,255\} //at most 256 characters

number ::= non\_zero\_digit{digit}  
float ::= (number|"0")."(digit{digit})

const\_num ::= ("+"|"-"?) (number|float) | "0"

const\_character ::= ""character""  
character ::= letter|digit|symbol

const\_string ::= \"string\"  
string ::= {character}

const\_bool ::= "true" | "false"

reserved\_words ::= "if"  
| "while"  
| "bool"  
| "char"  
| "int"  
| "string"  
| "float"

// Syntax

program ::= statement\_list  
statement\_list ::= statement | statement statement\_list  
statement ::= simple\_statement | compund\_statement

simple\_statement ::= (assignment\_statement  
| iostatement  
| declaration\_statement);"

compound\_statement ::= if\_statement | while\_statement

simple\_type ::= "bool"  
| "char"

- | "int"
- | "string"
- | "float"

array\_type ::= simple\_type["number"]

type ::= simple\_type | array\_type

constant ::= const\_num

- | const\_character
- | const\_string
- | const\_bool

expression ::= constant

- | identifier
- | identifier["number"]
- | expression binary\_operator expression
- | unary\_operator expression
- | "("expression")"

declaration\_statement ::= type identifier

- | type identifier="expression"

iostatement ::= ("<<"identifier) | (">>"expression)

assignment\_statement ::= identifier "=" expression

if\_statement ::= if "("expression")" "{"statement\_list}" ["else" "{"statement\_list}"]

while\_statement ::= while "("expression")" "{"statement\_list}"

Atom

-----  
identifier

constant

int

char

bool

string

float

>>

<<

while

if

else

and

or

!

+

-

\*

/

%

>  
<  
>=  
<=  
!=  
==  
=  
;  
[  
]  
{  
}  
(  
)  
.  
^

```
p1
#computes the maximum
int a=9;
int b=6;
if(a>b){
    >>"a is the maximum";
}else{
    >>"b is the maximum";
}
```

```
p2
#computes the gcd
int a=9;
int b=6;
while(a!=b){
    if (a>b){
        a=a-b;
    }
    if (a<b){
        b=b-a;
    }
}
>>a." is the gcd";
```

```
p3
#prints the square of the elements of an array
int[256] a;
int i=0;
int n;
<<n;
while (i<n){
    <<a[i];
    i=i+1;
}
i=0;
while (i<n){
    >>"square of",a[i]," is ",a[i]^2;
```

```
i=i+1;  
}
```

p4

```
1a=9;  
@b=6;  
if (a>b){  
  >>"a is the maximum"  
}else{  
  >>"b is the maximum"  
}
```