

MiniC Language Specification

Syntax

Keywords

B, C, I, U, IF, ELSE, FOR, WHILE, RET

Keywords are reserved and cannot be used as identifiers.

Comment

Comments are started by `//` and are terminated by the end of the line.

Whitespace

Tabs, spaces and newlines are considered as whitespace which separates tokens

Identifiers

Case Sensitive, Alphanumeric with starting character a letter.

Datatypes

Integer - 32 bit signed integer, declared with keyword I

Unsigned Integer - 32 bit unsigned integer, declared with keyword U

Character - any printable ASCII character, character literals must be enclosed in single quotes, declared with keyword C

Bool - Only two values 0 and 1, declared with keyword B

1D or 2D Array of any of the preceding types.

Arithmetic Operations

Addition, Subtraction, Multiplication, Division and Modulus are supported

Logical Operations

Logical AND, OR, NOT and Comparison operations (`>`, `>=`, `<`, `<=`, `==`, `!=`) are supported.

Semantics

Scope

There are two scopes, global - anything defined outside of a function which can be referenced anywhere and local - anything defined inside a function definition and can only be referenced inside the same function. If same identifier is defined in two scopes the local one takes precedence.

Assignment

A value can be assigned to a variable as `<var> = <expr>`, which copies the value of the expression to the location of the variable.

Control Statements

IF ELSE

```
IF (<expr>)  
{ <block> }  
  
ELSE  
  
{ <block>}
```

If the value of the expression is non zero then the first block is executed else the second block is executed.

FOR

```
FOR (iterator_definition,cond_expr,iterator_increment)  
{ block }
```

first the the definition is executed, then the cond_expr is evaluated and if non zero the block and subsequently the increment assignment is executed until the cond_expr evaluates to false or 0.

WHILE

```
WHILE ( <expr> )  
{ <block> }
```

The block is run as long as the expr evaluates to a non zero value.

Expressions

Follow normal left to right and bracket associative rules.

Operator precedence is !,*,/,+,-,>,>=,<,<=,==,!=,&,|

Semantic checks

All identifiers and functions must be defined before used.

No identifier may be defined more than once in same scope.

Both location of the assignement and the expression must be of same datatype.

The conditional expression in an IF,FOR or WHILE statement must have type boolean.

Operands for arithmetic should be of compatible type.