## 语言概述：

类型：int、string、bool、float.

支持数组。

支持函数。

支持 if-else 、while、for

支持注释

支持输入输出。（怎么弄。。。）

支持结构。

计算基本表达式。（与或非+-\*/<=>（）之类的）

## 正则定义

Identifier字母(下划线)开头，其余为数字或字母或下划线。

整数：

字符串：""。

操作符：&&、||、!、+、-、\*、/、<、=、<、<=、==、!=、>=。

其他： ， ； space \n （） {} []

关键字：if、else、while、for、int、string、float、void、return、true、false、struct、function

## 操作符优先级

!

\* /

+ -

> >= < <=

==

&&

||

## 产生式规则

program --->不知道怎么写

block --->{stmts}

stmts --->stmts stmt | empty

decl --->type id

type --->type[ num ] | basic

basic --->int | string

var --->var[num] | id

stmt --->decl ;

| var =expr ;

| if ( bool ) stmt

| if ( bool ) stmt else stmt

| while ( bool ) stmt

| break ;

| block

bool --->bool || lv7 | lv7

lv7 --->lv7 && lv6 | lv6

lv6 --->lv6 == lv5 |lv6 != lv5 | lv5

lv5 --->lv5 < expr | lv5 <= expr | lv5 > expr | lv5 >= expr | expr

expr --->expr + lv3 | expr - lv3 | lv3

lv3 --->lv3 \* lv2 | lv3 / lv2 | lv2

lv2 ---> ! lv1 | -lv1

lv1 --->(expr) | var | num | true | false

**function** 函数名**(**参数表**):**返回类型

参数表为空的话允许为void

不返回东西的话可以把返回类型写成void，也可以留空（在此种情况下省略冒号）

例子：

function calc(string expr):int {...}

function set(int value):void {...}

function play(int id, float time){...}