**Mini-Language Specification**

Alphabet

* A-Z, a-z (Uppercase and lowercase letters)
* 0-9 (Digits)
* \_

Operators

* +, -, \*, /, (Addition, Subtraction, Multiplication, Division)
* =, !=, >, <,<=,>=(Equality, Inequality, Greater, Lower,...)
* &&, ||, ! (And, Or, Not - Logical)
* <- (Assignment)

Separators

( ) { } [ ] space : ; ,

Keywords

* int
* string
* char
* array
* declare
* read
* write
* if
* else
* repeat
* until
* for
* from

Identifiers

* identifier = letter {letter|digit} | letter
* letter = “A”| “B”| …| “Z”| “a”| “b”| …|”z”
* digit = “0”| non\_zero\_digit
* non\_zero\_digit = “1”| … “9”

Constants

* int = “0” |[“+”|”-“] non\_zero\_digit {digit}
* char = ‘letter’|’digit’
* string = “{letter|digit}”

|  |
| --- |
| **Token** |
| + |
| - |
| \* |
| / |
| <- |
| = |
| != |
| < |
| > |
| <= |
| >= |
| { |
| } |
| [ |
| ] |
| ( |
| ) |
| : |
| ; |
| , |
|  |
| int |
| String |
| char |
| array |
| declare |
| read |
| write |
| if |
| else |
| repeat |
| until |
| for |
| from |

Syntax

* program = cmpdStmt
* cmpdStmt = "{" declarations statements "}"
* declarations = {declaration}
* declaration = "declare" IDENTIFIER ":" type ";"
* type = typeSimple | arrayType
* typeSimple = "int" | "string"
* arrayType = "array" "(" typeSimple ")"
* statements = {simpleStmt ";" | structStmt}
* stmt = simpleStmt | structStmt
* simpleStmt = assignStmt | ioStmt
* structStmt = ifStmt | loopStmt
* loopStmt = forStmt | untilStmt
* exp = term {("+" | "-") exp} | constant
* term = factor {("\*" | "/") term}
* factor = "(" exp ")" | IDENTIFIER | int
* RELATION = "<" | "<=" | "=" | "!=" | ">=" | ">"
* condition = exp RELATION exp
* assignStmt = IDENTIFIER "<-" exp
* ioStmt = ("read" | "write") "(" exp ")"
* ifStmt = "if" condition cmpdStmt ["else" compStmt]
* untilStmt = "repeat" cmpdStmt "until" condition
* forStmt = "for" IDENTIFIER "from" "(" exp "," exp "," exp ")" cmpdStmt
* constant = int | char | string | emptyArray
* emptyArray = "[" "]"

Examples

{

declare a:int;

declare b:int;

declare c:int;

declare max:int;

declare min:int;

a <- "aa aa";

b <- " ";

read(a);

read(b);

read(c);

if a>b

{

if b>c

{

max <- a;

min <- c;

}

else

{

min <- b;

if a>c

{max <- a;}

else

{max <- c;};

};

}

else

{

if b<c

{

min <- a;

max <- c;

}

else

{

max <- b;

if a > c

{min <- c;}

else

{min <- a;};

};

};

write(min);

write(max);

}