

Lesther Kevin Federico López Miculax

Carnet: 202110897

%lex

%options case-insensitive

%x string

int (?:[0-9]|[1-9][0-9]+)

frac (?:\.[0-9]+)

%%

"," return 'PTCOMA';

"(" return 'PARIZQ';

")" return 'PARDER';

"." return 'PUNTO';

":" return 'DOSPUNTOS';

"," return 'COMA';

"[" return 'CORIZR';

"]" return 'CORDER';

"{" return 'LLAVEIZQ';

"}" return "LLAVEDER";

"?" return 'KLEENE';

"<=" return "MENOROIGUALQUE";

">=" return 'MAYOROIGUALQUE';

"==" return "IGUALACION";

"=" return 'IGUAL';

"\$" return 'DOLAR';

"++" return 'AUMENTO';

"--" return 'REDUCCION';

"!="	return 'DIFERENCIACION';
"<"	return "MENORQUE";
">"	return 'MAYORQUE';
"!"	return 'NOT';
"&&"	return 'AND';
" "	return 'OR';
"print"	return 'RPRIN';
"true"	return 'TRUE';
"false"	return 'FALSE';
"if"	return 'IF';
"while"	return 'WHILE';
"else"	return 'ELSE';
"void"	return 'VOID';
"toLower"	return 'TOLOWER';
"toUpper"	return 'TOUPPER';
"return"	return 'RRETURN';
"new"	return 'NEW';
"list"	return 'LIST';
"add"	return 'ADD';
"switch"	return 'SWITCH';
"case"	return 'CASE';
"default"	return 'DEFAULT';
"for"	return 'FOR';
"do"	return 'DO';
"break"	return 'BREAK';
"continue"	return 'CONTINUE';
"Length"	return 'LENGTH';
"truncate"	return 'TRUNCATE';

```

"round"      return 'ROUND';
"typeof"     return 'TYPEOF';
"toString"   return 'TOSTRING';
"toCharArray" return 'TOCHARARRAY';
"main"       return 'MAIN';
"+"          return 'MAS';
"-"          return 'MENOS';
"*"          return 'POR';
"/"          return 'DIVISION';
"^"          return 'POTENCIA';
"%"          return 'MODULO';
"int"        return 'ENTERO';
"string"     return 'RSTRING';
"char"       return 'RCHAR';
"boolean"    return 'RBOOLEAN';
"double"     return 'RDOUBLE';

[ \r\t]+
\n

"/".*

[/][^]*[*]+([^\/*][^]*[*]+)*[/]

[a-zA-Z][a-zA-Z0-9_ñÑ]* return 'ID';

{int}{frac}\b return 'DECIMAL';

[0-9]+\b return 'ENTERO';

\'((\\\'|^[^\\\'])*\'

["]

<string>[^"\\]+

<string>"\\\\"

<string>"\\n"

<string>"\\t"

```

<string>"\\\\"

<string>"\\\'

<string>["]

/\[^\"]*\\"

%left 'AUMENTO' 'REDUCCION'

%left 'POTENCIA'

%left 'OR'

%left 'AND'

%left 'DIFERENCIACION' 'MENORQUE' 'MAYORQUE' 'MENOROIGUALQUE' 'MAYOROIGUALQUE'
'IGUALACION'

%left 'MAS' 'MENOS'

%left 'POR' 'DIVISION' 'MODULO'

%right 'UMENOS'

%right 'NOT'

%start INICIO

%%

INICIO

: INSTRUCCIONES EOF

;

INSTRUCCIONES

: INSTRUCCIONES INSTRUCCION

| INSTRUCCION

;

TIPO

: RDOUBLE

- | RENTERO
- | RCHAR
- | RSTRING
- | RBOOLEAN
- | VOID

;

BLOQUE_INSTRUCCIONES

: LLAVEIZQ INSTRUCCIONES LLAVEDER

- | LLAVEIZQ LLAVEDER

;

INSTRUCCION

: DEFPRINT

- | DECLARAR
- | DIF
- | FUNCION
- | DWHILE
- | LLAMAR_FUNCION PTCOMA
- | DFOR
- | DOWHILE
- | SWITCHCASE
- | INCREMENTOYDECREMENTO PTCOMA
- | DTOLOWER PTCOMA
- | DTOUPPER PTCOMA
- | VECTOR
- | MODIFICARVECTOR
- | DRETURN

```

| CONTINUE PTCOMA
| BREAK PTCOMA
| LISTA
| MAIN LLAMAR_FUNCION PTCOMA
| AGREGARVALORLISTA
| MODIFICARVALORLISTA
| ASIGNACION
| error
;
DEFPRINT
: RPRIN PARIZQ EXPRESION PARDER PTCOMA
;
DECLARAR
: TIPO ID PTCOMA
| TIPO ID IGUAL EXPRESION PTCOMA
;
DRETURN
: RRETURN PTCOMA
| RRETURN EXPRESION PTCOMA
;
ASIGNACION
: ID IGUAL EXPRESION PTCOMA }
;
DIF
: IF PARIZQ EXPRESION PARDER BLOQUE_INSTRUCCIONES
| IF PARIZQ EXPRESION PARDER BLOQUE_INSTRUCCIONES DELSE
;
DELSE

```

```
: ELSE DIF }  
| ELSE BLOQUE_INSTRUCCIONES  
;  
FUNCION  
: TIPO ID PARIZQ LISTA_PARAMETROS PARDER BLOQUE_INSTRUCCIONES  
| TIPO ID PARIZQ PARDER BLOQUE_INSTRUCCIONES  
  
;
```

```
LISTA_PARAMETROS  
: LISTA_PARAMETROS COMA PARAMETRO  
| PARAMETRO  
;
```

```
PARAMETRO  
: TIPO ID  
;  
DWHILE  
: WHILE PARIZQ EXPRESION PARDER BLOQUE_INSTRUCCIONES  
;
```

```
DFOR  
: FOR PARIZQ CONDICIONINICIALFOR EXPRESION PTCOMA EXPRESION PARDER  
BLOQUE_INSTRUCCIONES  
;
```

```
CONDICIONINICIALFOR  
: ASIGNACION  
| DECLARAR
```

;

DOWHILE

: DO BLOQUE_INSTRUCCIONES WHILE PARIZQ EXPRESION PARDER PTCOMA

;

LISTA_EXPRESIONES

: LISTA_EXPRESIONES COMA EXPRESION

;

VECTOR

: TIPO CORIZR CORDER ID IGUAL NEW TIPO CORIZR ENTERO CORDER PTCOMA

| TIPO CORIZR CORDER ID IGUAL LLAVEIZQ LISTA_EXPRESIONES LLAVEDER PTCOMA

;

MODIFICARVECTOR

: ACCEDERVECTOR IGUAL EXPRESION PTCOMA

;

LISTA

: LIST MENORQUE TIPO MAYORQUE ID IGUAL NEW LIST MENORQUE TIPO MAYORQUE PTCOMA

;

AGREGARVALORLISTA

: ACCEDERID PUNTO PARIZQ EXPRESION PARDER PTCOMA

;

MODIFICARVALORLISTA

: ACCEDERLISTA IGUAL EXPRESION PTCOMA

;

EXPRESION

: PRIMITIVO

| EXPRESION MAS EXPRESION

| EXPRESION MENOS EXPRESION

| EXPRESION DIVISION EXPRESION

| EXPRESION POR EXPRESION

| MENOS EXPRESION %prec U MENOS

| EXPRESION POTENCIA EXPRESION

| EXPRESION MODULO EXPRESION

| ACCEDERID

| LLAMAR_FUNCION

| PARIZQ EXPRESION PARDER

| EXPRESION AND EXPRESION

| EXPRESION OR EXPRESION

| NOT EXPRESION

| EXPRESION IGUALACION EXPRESION

| EXPRESION DIFERENCIACION EXPRESION

| EXPRESION MENORQUE EXPRESION

| EXPRESION MENOROIGUALQUE EXPRESION

| EXPRESION MAYORQUE EXPRESION

| EXPRESION MAYOROIGUALQUE EXPRESION

| INCREMENTOYDECREMENTO

| DTOLOWER

| DTOUPPER

| DLENGTH

| DTRUNCATE

| DROUND

| DTYPEOF

| DSTOSTRING

```
| ACCEDERVECTOR
| ACCEDERLISTA
| OPERADORTERNARIO
| DIF
| DFOR
| DWHILE
| WHILE
;
LLAMAR_FUNCION
: ID PARIZQ LISTA_EXPRESIONES PARDER
| ID PARIZQ PARDER
;
DTOLOWER
: TOLOWER PARIZQ EXPRESION PARDER
;
DTRUNCATE
: TRUNCATE PARIZQ EXPRESION PARDER
;
DLENGTH
: LENGTH PARIZQ EXPRESION PARDER
;
DTOUPPER
: TOUPPER PARIZQ EXPRESION PARDER
;
DROUND
: ROUND PARIZQ EXPRESION PARDER
;
DTYPEOF
```

```
: TYPEOF PARIZQ EXPRESION PARDER
;
DTOSTRING
: TOSTRING PARIZQ EXPRESION PARDER  }
;
ACCEDERLISTA
: ACCEDERID CORIZR CORIZR EXPRESION CORDER CORDER
;
ACCEDERVECTOR
: ACCEDERID LLAVEIZQ EXPRESION LLAVEDER
;
```

```
ACCEDERID
: ID
;
//INCREMENTO Y DECREMENTO
INCREMENTOYDECREMENTO
: ID AUMENTO
| ID REDUCCION
;
```

```
PRIMITIVO
: DECIMAL
| ENTERO
| CADENA
| CHARACTER
| TRUE
| FALSE
;
```

