Name : Sabadus Alex

Git: <https://github.com/SabaAlex/FLCD/blob/master/Lab%209/my_toy_language.y>

%{

#include <stdio.h>

#include <stdlib.h>

#define YYDEBUG 1

#define TIP\_INT 1

#define TIP\_CAR 2

#define TIP\_STRING 3

double stiva[20];

int sp;

void push(double x)

{ stiva[sp++]=x; }

double pop()

{ return stiva[--sp]; }

%}

%union {

int l\_val;

char \*p\_val;

}

%token ELSE

%token IF

%token OUTPUT

%token INPUT

%token VAR

%token FOR

%token CHAR

%token NUMBER

%token STRING

%token TYPE

%token ARRAYTYPE

%token NE

%token LE

%token GE

%token EQ

%token LT

%token GT

%token ID

%token <p\_val> CONST\_INT

%token <p\_val> CONST\_CHAR

%token <p\_val> CONST\_STRING

%token ATRIB

%left '+' '-'

%left DIV MOD '\*' '/'

%left OR

%left AND

%type <l\_val> constantaNum

%type <p\_val> constantaSir

%%

cmpdStmt: '{' stmtlist '}'

;

program: cmpdStmt

;

stmtlist: stmt

| stmt stmtlist

;

structStmt: ifStmt

| forStmt

| cmpdStmt

;

stmt: simplStmt ';'

| declList ':'

| structStmt

;

ioStmt: INPUT

| OUTPUT ID

| OUTPUT constanta

;

simplStmt: assignStmt

| ioStmt

;

assignStmt: ID '=' expression

expression: expression '+' term

| expression '-' term

| term

;

term: term '\*' factor

| term '/' factor

| term '%' factor

| factor

;

constanta: constantaNum

| constantaSir

;

factor: '(' expression ')'

| ID

| constanta

;

declList: VAR declaration lista\_var

| VAR declareAndAsignStmt lista\_var

;

lista\_var: ',' declaration

| ',' declareAndAsignStmt

;

declareAndAsignStmt: declaration "=" expression

;

declaration: ID ':' type

;

type: typeSimple

| arrayDecl

;

typeSimple: STRING

| CHAR

| NUMBER

;

constantaNum: CONST\_INT {

$$=TIP\_INT;

push(atof($1));

}

;

arrayDecl: typeSimple '.' '.' '.' '(' constantaNum ')'

;

constantaSir: CONST\_CHAR {

$$ = TIP\_CAR;

push((char)$1[1]);

}

| CONST\_STRING {

$$ = TIP\_STRING;

push($1);

}

;

ifStmt: IF '(' condition ')' stmt ELSE stmt

| IF '(' condition ')' stmt ELSE cmpdStmt

| IF '(' condition ')' cmpdStmt ELSE stmt

| IF '(' condition ')' cmpdStmt ELSE cmpdStmt

;

forStmt: FOR '(' assignStmt condition ';' assignStmt ')' stmt

| FOR '(' assignStmt condition ';' assignStmt ')' cmpdStmt

| FOR '(' declList condition ';' assignStmt ')' stmt

| FOR '(' declList condition ';' assignStmt ')' cmpdStmt

;

condition: expression RELATION expression

;

RELATION: LT

| LE

| EQ

| NE

| GE

| GT

| AND

| OR

;

%%

yyerror(char \*s)

{

printf("%s\n", s);

}

extern FILE \*yyin;

main(int argc, char \*\*argv)

{

if(argc>1) yyin = fopen(argv[1], "r");

if((argc>2)&&(!strcmp(argv[2],"-d"))) yydebug = 1;

if(!yyparse()) fprintf(stderr,"\tO.K.\n");

}