%{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

YACC file

基础程序

Date:2023/9/19

forked SherryXiye

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

#ifndef YYSTYPE

#define YYSTYPE double

#endif

int yylex();

int lineno = 1;

int tokenval = 0 ;

char strval [20];

char idlist [100][20];

int idnum = 0;//id的总数

int idnow = 1 ;

int idleft = 1;

int idval[100];

int vl=0;//记录是否是左边

int equalid [100];//记录全部等待赋值的id

int equalnum = 0 ;//记录等待赋值的id的数量

extern int yyparse();

FILE\* yyin;

void yyerror(const char\* s);

%}

%token PLUS MINUS TIMES DIVIDE LPAREN RPAREN INTEGER ID EQUAL

//注意先后定义的优先级区别

%left PLUS MINUS

%left TIMES DIVIDE

%right UMINUS

%%

lines : lines expr '\n' {printf("%.2f",$2) ;}

| lines '\n'

|

;

expr : expr PLUS expr { $$=$1+$3; }

| expr MINUS expr { $$=$1-$3;}

| expr TIMES expr { $$=$1\*$3;}

| expr DIVIDE expr { $$=$1/$3;}

| LPAREN expr RPAREN { $$=$2;}

| MINUS expr %prec UMINUS {$$=-$2;}

| LEFTVARIABLE EQUAL expr {$$=$3;idval[equalid[equalnum-1]]=$3;equalnum-=1;}

| NUMBER

| VARIABLE

;

LEFTVARIABLE: VARIABLE {equalid[equalnum] = idnow; equalnum +=1;}

;

;

VARIABLE: ID {$$=idval[idnow];}

;

NUMBER : INTEGER {$$=tokenval;}

;

%%

// programs section

int yylex()

{

int t;

while (1) {

t = getchar();

if (t == ' ' || t == '\t')

;

else if (t == '\n')

return '\n';

else if (isdigit(t)) {

tokenval = 0;

while (isdigit(t)) {

tokenval = tokenval \* 10 + t - '0';

t = getchar();

}

ungetc(t, stdin);

return INTEGER;

}

else if (isalpha(t)){

strval[0]='\0';

while(isdigit(t) || isalpha(t)){

char temp[2] = {t,'\0'};

strcat(strval,temp);

t = getchar();

}

for(idnow = 1;idnow<=idnum;idnow++){

if(strcmp(idlist[idnow],strval)==0){

break;

}

}

strcpy(idlist[idnow],strval);

if(idnow == idnum+1 ){

idnum += 1;

idnow = idnum ;

}

ungetc(t,stdin);

return ID;

}

else {

switch(t) {

case '+': return PLUS;

case '-': return MINUS;

case '\*': return TIMES;

case '/': return DIVIDE;

case '(': return LPAREN;

case ')': return RPAREN;

case '=': return EQUAL;

default:

tokenval = 0;

return t;

}

}

}

}

int main(void)

{

yyin=stdin;

do{

yyparse();

}while(!feof(yyin));

return 0;

}

void yyerror(const char\* s){

fprintf(stderr,"Parse error: %s\n",s);

exit(1);

}