%{

#include<stdio.h>

#include<string.h>

struct quad

{

char op[5];

char arg1[10];

char arg2[10];

char result[10];

}QUAD[30];

int i=0,j;

%}

%union

{

char exp[10];

}

%token <exp> VAR

%type <exp> S E T F

%%

S: E { printf("\n\n\tThere are %d quadrupls",i);

printf("\n\tList of Quadruples :\n");

for(j=0;j<i;j++)

printf("\t%s\t%s\t%s\t%s\n",QUAD[j].op,QUAD[j].arg1,QUAD[j].arg2,QUAD[j].result);

}

;

E: E'+'T { printf("\n E->E+T \t:$$=%s, $1=%s, $3=%s",$$,$1,$3);

strcpy(QUAD[i].op,"+");

strcpy(QUAD[i].arg1,$1);

strcpy(QUAD[i].arg2,$3);

strcpy(QUAD[i].result,$$);

i++;

}

| T { printf("\n E->T \t\t:$$=%s, $1=%s",$$,$1);}

;

T: T'\*'F { printf("\n T->T\*F \t:$$=%s, $1=%s, $3=%s",$$,$1,$3);

strcpy(QUAD[i].op,"\*");

strcpy(QUAD[i].arg1,$1);

strcpy(QUAD[i].arg2,$3);

strcpy(QUAD[i].result,$$);

i++;

}

| F { printf("\n T->F \t\t:$$=%s, $1=%s",$$,$1);}

;

F: VAR {printf("\n F->VAR \t:$$=%s, $1=%s",$$,$1);}

;

%%

main()

{

yyparse();

}

void yyerror(char\* s)

{

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

}