|  |
| --- |
| ALPHA [A-Za-z] |
|  | DIGIT [0-9] |
|  | %% |
|  | if return IF; |
|  | then return THEN; |
|  | else return ELSE; |
|  | {ALPHA}({ALPHA}|{DIGIT})\* return ID; |
|  | {DIGIT}+ {yylval=atoi(yytext); return NUM;} |
|  | [ \t] ; |
|  | \n yyterminate(); |
|  | . return yytext[0]; |
|  | %% |

Yacc file

|  |
| --- |
| %token ID NUM IF THEN ELSE |
|  | %right '=' |
|  | %left '+' '-' |
|  | %left '\*' '/' |
|  | %left UMINUS |
|  | %% |
|  |  |
|  | S : IF '(' Y ')'{lab();} THEN '{' X '}'{lab2();} ELSE '{' X '}' {lab3();} |
|  | ; |
|  | X : E ';'|X X; |
|  | Y : B {abc();codegen\_assigna();first();} |
|  | | B '&''&'{abc();codegen\_assigna();second();} Y |
|  | | B {abc();codegen\_assigna();third();}'|''|' Y |
|  | | '!'B{abcde();codegen\_assigna();first();} |
|  | ; |
|  |  |
|  | B : V '='{push();}'='{push();}D |
|  | | V '>'{push();}F |
|  | | V '<'{push();}F |
|  | | V '!'{push();}'='{push();}D |
|  | |'(' B ')' |
|  | | V{pushab();} |
|  |  |
|  | ; |
|  | F :'='{push();}D |
|  | |D{pusha();} |
|  | ; |
|  | D :NUM{push();} |
|  | |ID{push();} |
|  | ; |
|  | E :V '='{push();} E{codegen\_assign();} |
|  | | E '+'{push();} E{codegen();} |
|  | | E '-'{push();} E{codegen();} |
|  | | E '\*'{push();} E{codegen();} |
|  | | E '/'{push();} E{codegen();} |
|  | | '(' E ')' |
|  | | '-'{push();} E{codegen\_umin();} %prec UMINUS |
|  | | V |
|  | | NUM{push();} |
|  | | S |
|  | ; |
|  | V : ID {push();} |
|  | ; |
|  | %% |
|  |  |
|  | #include "lex.yy.c" |
|  | #include<ctype.h> |
|  | char st[100][10]; |
|  | int top=0; |
|  | char i\_[2]="0"; |
|  | char temp[2]="t"; |
|  | int abcd=0; |
|  | int label[20]; |
|  | int lnum=0; |
|  | int ltop=0; |
|  | int i=0; |
|  | main() |
|  | { |
|  | printf("Enter the expression : "); |
|  | yyparse(); |
|  | } |
|  |  |
|  | pusha() |
|  | { |
|  | strcpy(st[++top]," " ); |
|  | } |
|  | pushab() |
|  | { |
|  | strcpy(st[++top]," "); |
|  | strcpy(st[++top]," "); |
|  | strcpy(st[++top]," "); |
|  | } |
|  | push() |
|  | { |
|  | strcpy(st[++top],yytext); |
|  | } |
|  |  |
|  | abc() |
|  | { |
|  | abcd++; |
|  | printf("\nX%d : if ",abcd); |
|  | } |
|  | abcde() |
|  | { |
|  | abcd++; |
|  | printf("\nX%d :not ",abcd); |
|  | } |
|  | second1() |
|  | { |
|  | printf("\nif x%d true goto L%d\n",abcd,lnum); |
|  | printf("\nif x%d false goto L%d\n",abcd,++lnum); |
|  | lnum=lnum-1; |
|  | } |
|  | second() |
|  | { |
|  | int xyz=0; |
|  | xyz=abcd+1; |
|  | printf("falg=true else flag=false"); |
|  | printf("\n if flag(true) goto x%d",xyz); |
|  | printf("\n if flag(false) goto L1"); |
|  | } |
|  | first() |
|  | { |
|  | printf("flag=true else flag=false"); |
|  | printf("\n if flag(true) goto L0"); |
|  | printf("\n if flag(false) goto L1"); |
|  | } |
|  | third() |
|  | { |
|  | int xyz=0; |
|  | xyz=abcd+1; |
|  | printf("flag=true else flag=false"); |
|  | printf("\n if flag(true) goto L0 "); |
|  | printf("\n if flag(false) goto x%d",xyz); |
|  | } |
|  | codegen() |
|  | { |
|  | strcpy(temp,"t"); |
|  | strcat(temp,i\_); |
|  | printf("%s = %s %s %s\n",temp,st[top-2],st[top-1],st[top]); |
|  | top-=2; |
|  | strcpy(st[top],temp); |
|  | i\_[0]++; |
|  | } |
|  |  |
|  | codegen\_umin() |
|  | { |
|  | strcpy(temp,"t"); |
|  | strcat(temp,i\_); |
|  | printf("%s = -%s\n",temp,st[top]); |
|  | top--; |
|  | strcpy(st[top],temp); |
|  | i\_[0]++; |
|  | } |
|  |  |
|  | codegen\_assigna() |
|  | { |
|  | printf("%s %s %s %s ",st[top-3],st[top-2],st[top-1],st[top]); |
|  | top-=3; |
|  | } |
|  | codegen\_assign() |
|  | { |
|  | printf("%s = %s\n",st[top-2],st[top]); |
|  | top-=2; |
|  | } |
|  | codegen\_assignb() |
|  | { |
|  | printf("%s %s %s ",st[top-3],st[top-2],st[top-1]); |
|  | top-=3; |
|  | } |
|  |  |
|  | lab() |
|  | { |
|  | printf("\nL0 :\n"); |
|  | } |
|  | lab1() |
|  | { |
|  | strcpy(temp,"t"); |
|  | strcat(temp,i\_); |
|  | printf("\n%s = not arguement \n",temp); |
|  | printf("if %s goto L%d\n",temp,lnum); |
|  | i\_[0]++; |
|  | label[++ltop]=lnum; |
|  | } |
|  |  |
|  | lab2() |
|  | { |
|  | int x; |
|  | lnum++; |
|  | x=label[ltop--]; |
|  | printf("goto L2\n"); |
|  | printf("L%d: \n",++x); |
|  | label[++ltop]=lnum; |
|  | } |
|  |  |
|  | lab3() |
|  | { |
|  | int y; |
|  | y=label[ltop--]; |
|  | printf("L2: \n"); |
|  | } |