```
%{
#include<string.h>
%}
%token ID NUM WHILE
%right '='
%left '+' '-'
%left '*' '/'
%left GE LE '<' '>'
%%

S : WHILE{lab1();} '(' E ')'{lab2();} E ';'{lab3();}
  ;
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();}
  | E '<'{push();} E{codegen();}
  | E GE {push();} E{codegen();}
  | E LE {push();} E{codegen();}
  | '(' E ')'
  | V
  | NUM{push();}
  ;
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 lnum=0;
int start=0;
main()
 {
 printf("Enter the expression : ");
 yyparse();
 }




push()
 {
  strcpy(st[++top],yytext);
 }

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_assign()
 {
 printf("%s = %s\n",st[top-2],st[top]);
 top-=2;
 }

lab1()
{
printf("L%d: \n",lnum++);
}

lab2()
{
 strcpy(temp,"t");
 strcat(temp,i_);
 printf("%s = not %s\n",temp,st[top]);
 printf("if %s goto L%d\n",temp,lnum);
 i_[0]++;
 }

lab3()
{
printf("goto L%d \n",start);
printf("L%d: \n",lnum);
}
int yyerror(char *s)
 {
    printf("%s\n", s);
}
```

```
/output
sweta@sweta-HP-Pavilion-dv6-Notebook-PC:~/Desktop$ lex icg.l
sweta@sweta-HP-Pavilion-dv6-Notebook-PC:~/Desktop$ yacc icg.y
sweta@sweta-HP-Pavilion-dv6-Notebook-PC:~/Desktop$ gcc y.tab.c -ll
sweta@sweta-HP-Pavilion-dv6-Notebook-PC:~/Desktop$ ./a.out
Enter the expression : while(i>=0) c=c-d;
L0:
t0 = i >= 0
t1 = not t0
if t1 goto L1
t2 = c - d
```

```
c = t2
goto L0
L1:
```

icg.l

```
ALPHA [A-Za-z]
DIGIT [0-9]
%%
while                return WHILE;
{ALPHA}({ALPHA}|{DIGIT})*    return ID;
{DIGIT}+            {yylval=atoi(yytext); return NUM;}
">="                return GE;
"<="                return LE;
[ \t]               ;
\n                yyterminate();
.                 return yytext[0];
%%
```

icg.y

```
%{
#include<string.h>
%}
%token ID NUM WHILE
%right '='
%left '+' '-'
```

```
%left '*' '/'
%left GE LE '<' '>'
%%

S : WHILE{lab1();} '(' E ')'{lab2();} E ';'{lab3();}
  ;
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();}
  | E '<'{push();} E{codegen();}
  | E GE {push();} E{codegen();}
  | E LE {push();} E{codegen();}
  | '(' E ')'
  | V
  | NUM{push();}
  ;
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 lnum=0;
int start=0;
main()
 {
 printf("Enter the expression : ");
 yyparse();
 }




push()
 {
  strcpy(st[++top],yytext);
 }

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_assign()
```

```
 {
 printf("%s = %s\n",st[top-2],st[top]);
 top-=2;
 }

lab1()
{
printf("L%d: \n",lnum++);
}

lab2()
{
 strcpy(temp,"t");
 strcat(temp,i_);
 printf("%s = not %s\n",temp,st[top]);
 printf("if %s goto L%d\n",temp,lnum);
 i_[0]++;
 }

lab3()
{
printf("goto L%d \n",start);
printf("L%d: \n",lnum);
}
int yyerror(char *s)
 {
    printf("%s\n", s);
}
```

output:

```
[root@localhost ~]# lex icg3.l

[root@localhost ~]# yacc icg3.y

[root@localhost ~]# gcc y.tab.c -ll

[root@localhost ~]# ./a.out

Enter the expression : while(i>=0) c=c-d;

L0:

t0 = i >= 0

t1 = not t0

if t1 goto L1

t2 = c - d

c = t2

goto L0

L1:

[root@localhost ~]#
```