```
%{
#include "sample.tab.h"
extern int yyerror(char *str);
extern int yyparse();
%}
%%
"while" return WH;
"if" return IF;
"do" return DO;
"for" return FOR;
"(" return OP;
")" return CP;
"{" return OCB;
"}" return CCB;
"<" |
">" |
"<=" |
">=" |
"==" |
"!=" return CMP;
"+" |
"-" |
"*" |
"/" return OPR;
"=" return ASG;
([a-zA-Z])("_"|[a-zA-Z0-9])* return ID;
[0-9]+ return NUM;
";" return SC;
"," return COMMA;
" " {}
%%
int yywrap()
{
return 1;
}
```

```
%{
#include<stdio.h>
extern int yylex();
extern int yywrap();
extern int yyparse();
%}
%token WH IF DO FOR OP CP OCB CCB CMP SC ASG ID NUM COMMA OPR
%%

start: swh | mwh | dowh | sif | mif;
swh: WH OP cmplst CP stmt {printf("VALID SINGLE STATEMENT WHILE LOOP\n");};
mwh: WH OP cmplst CP OCB stlst CCB {printf("VALID MULTI STATEMENT WHILE LOOP\n");};
dowh: DO OCB stlst CCB WH OP cmplst CP SC {printf("VALID DO-WHILE LOOP\n");};
sif: IF OP cmplst CP stmt {printf("VALID SINGLE STATEMENT IF\n");};
```

```
mif: IF OP cmplst CP OCB stlst CCB {printf("VALID MULTI STATEMENT IF\n");};
cmplst: cmpn COMMA cmplst | cmpn ;
cmpn: ID CMP ID | ID CMP NUM;
stlst: stmt stlst | stmt;
stmt: ID ASG ID OPR ID SC | ID ASG ID OPR NUM SC | ID ASG NUM OPR ID SC | ID ASG NUM
OPR
NUM SC | ID ASG ID SC | ID ASG NUM SC
| start {printf("NESTED INSIDE A ");}
;
%%
int yyerror(char *str)
{
printf("%s", str);
}
main()
{
yyparse();
}
```