NAME : JOY MOJUMDAR
ID:CSE 066 07780

1 no.

Lex file:

```
%{
#include "y.tab.h"
#include <math.h>
extern double vbltable[26];
%}
%%
([0-9]+|([0-9]*\.[0-9]+)([eE][-+]?[0-9]+)?) {
yylval.dval = atof(yytext); return NUMBER;
}
[ \t] ; /* ignore whitespace */
[a-z] { yylval.vblno = yytext[0] - 'a'; return NAME; }
(min) {return MINIMUM;}
(floor) {return FLOOR;}
(log) {return LOG;}
(area) {return AREA;}
\n|. return yytext[0];
%%
```

Yac file:

```
%{
#include<stdio.h>
#include<stdlib.h>
#include <math.h>
extern FILE *yyin;
double vbltable[26] ;
```

NAME : JOY MOJUMDAR
ID:CSE 066 07780

```
void yyerror(const char *c){
fprintf(stderr,"%s",c);
}
int yylex();
%}
%union{
double dval;
int vblno;
}
%token LOG MINIMUM AREA FLOOR
%token <vblno> NAME
%token <dval> NUMBER
%type <dval> expression term factor expr

%%
statement_list: statement_list statement ';' '\n'
              | statement ';' '\n';
statement: NAME '=' expression { vbltable[$1] = $3; printf("%c = %lf\n",$1+'a',$3);}
           | expression { printf("= %g\n",$1); };
expression: expression '+' term { $$ = $1 + $3; }
          | expression '-' term { $$ = $1 - $3; }
           | term { };
term:MINIMUM '('expr','expr','expr')' {
if ($3 <$5 && $3< $7){
$$ = $3;
}
else if ( $5 < $3 && $5 < $7){
$$=$5;
}
```

NAME : JOY MOJUMDAR
ID:CSE 066 07780

```
else{
$$ = $7;
}
}
| expr { };

expr: FLOOR '('factor')' {$$ = floor($3);}
        | LOG '('factor')' {$$ = log10($3);}
        | AREA '('factor'-'factor')' { $$ = 3.1416 * ($3-$5) * ($3-$5);}
    | factor { }
;

factor: '-' factor { $$ = -$2; }
    | '(' expression ')' { $$ = $2; }
    | NUMBER {$$ = $1;}
    | NAME { $$ = vbltable[$1]; };

%%

int main(){
FILE *file;
file = fopen("code.c", "r") ;
        if (!file) {
        printf("Could not open file");
        exit (1);
        }
        else {
        yyin = file;
        }
```

NAME : JOY MOJUMDAR
ID:CSE 066 07780

yyparse();

}

2 no.

Lex file:

```
%{

#include "y.tab.h"

extern int lexval;

%}

%%

"do" { return DO;}

"while" {return WHILE;}

"print" {return PRINT;}

"to" { return TO;}

[0-9]+ {yylval = atoi(yytext); return NUMBER;}

[a-z] {yylval = atoi(yytext-'a');return TOKEN;}

[ \t]+ {}

'\n' {return 0;}
```

NAME : JOY MOJUMDAR
ID:CSE 066 07780

```
. {return yytext[0];}


%%
```

Yac file:

```
%{
#include <stdio.h>
#include <stdlib.h>
#include<math.h>
extern FILE *yyin;
int yylex();
int count_start;
int count_end;
int var = 0;
int increment;
%}

%token TOKEN PRINT WHILE DO BY NUMBER TO
%%
dowhile_statement :  do_statement while_cond ';' {};
do_statement : token'{'statement ';' '}' {};
while_cond :  token '(' expression')' {};
expression : TOKEN '=' expr TO expr BY expr{count_start = $3; count_end = $5; var = $1; increment =
$7;}
| expr{}
;
statement : command TOKEN {if(var == $2){
        var = count_start;
```

NAME : JOY MOJUMDAR
ID:CSE 066 07780

```
            do{
              printf("%d ", var);
              var += increment;
            }
            while(var <= count_end);
}};
expr: NUMBER '+' NUMBER { $$ = $1 + $3;}
   | NUMBER '-' NUMBER { $$ = $1 - $3;}
   | NUMBER '*' NUMBER { $$ = $1 * $3;}
   | NUMBER '/' NUMBER { $$ = $1 / $3;}
   | NUMBER { $$ = $1;};
command : PRINT{};
token : WHILE {}
     | DO {}
;
%%
int main(){
            FILE *file;
file = fopen("code.c", "r");
if(!file){
            printf("couldn't open file");
            exit(1);
}
else{
            yyin = file;
}
yyparse();
}
```