```
%{
        #include <stdio.h>
        #include <string.h>

        int installID(char *id);
        int installNum(char *num);

        char *ID_TABLE[1000];
        char *NUM_TABLE[1000];

        int total_id = 0;
        int total_num = 0;

%}

/* regular definitions */

delim [ \t\n]
ws {delim}+
letter [A-Za-z]
digit [0-9]
id      {letter}({letter}|{digit})*
number {digit}+(\.{digit}+)?(E[+-]?{digit}+)?

%%

{ws}    {

        fprintf(yyout, "Lexeme : WHITESPACE\nToken Name : NONE\nAttribute Value : NONE\n\n");

}
if              {

        fprintf(yyout, "Lexeme : %s\nToken Name : %s\nAttribute Value : NONE\n\n", yytext, yytext);

}
then    {

        fprintf(yyout, "Lexeme : %s\nToken Name : %s\nAttribute Value : NONE\n\n", yytext, yytext);

}
else    {

        fprintf(yyout, "Lexeme : %s\nToken Name : %s\nAttribute Value : NONE\n\n", yytext, yytext);

}
{id}    {

        char *str = yytext;
        int att_value = installID(str);
        fprintf(yyout, "Lexeme : %s\nToken Name : ID\nAttribute Value : %d\n\n", yytext, att_value);

}

{number}        {

        int att_value = installNum(yytext);
        fprintf(yyout, "Lexeme : %s\nToken Name : NUMBER\nAttribute Value : %d\n\n", yytext,
att_value);

}

"<"             {

        fprintf(yyout, "Lexeme : %s\nToken Name : RELOP\nAttribute Value : LT\n\n", yytext);

}

"<="    {

        fprintf(yyout, "Lexeme : %s\nToken Name : RELOP\nAttribute Value : LE\n\n", yytext);
```

```
}

"="             {

        fprintf(yyout, "Lexeme : %s\nToken Name : RELOP\nAttribute Value : EQ\n\n", yytext);

}

"<>"    {

        fprintf(yyout, "Lexeme : %s\nToken Name : RELOP\nAttribute Value : NE\n\n", yytext);

}

">"             {

        fprintf(yyout, "Lexeme : %s\nToken Name : RELOP\nAttribute Value : GT\n\n", yytext);

}

">="    {

        fprintf(yyout, "Lexeme : %s\nToken Name : RELOP\nAttribute Value : GE\n\n", yytext);

}

. {}

%%

int installID(char *id)
{
        int token_val = 0;
        if(total_id == 0)
        {
                ID_TABLE[0] = id;
                total_id = total_id + 1;
        }
        else
        {
                int found = 0;  //true - 1, false - 0
                int i;
                for(i=0; i< total_id; i++)
                {
                        if(strcmp(ID_TABLE[i], id) == 0)
                        {
                                found = 1;
                                token_val = i;
                                break;
                        }
                }
                if(found == 0)  //not found
                {
                        token_val = total_id;
                        ID_TABLE[total_id] = id;
                        total_id = total_id + 1;
                }
        }
        return token_val;
}


int installNum(char *num)
{
        int i;
        for(i=0; i<total_num; i++)
        {
                if(NUM_TABLE[i] == num)
                {
                        return i;
                }
        }
        NUM_TABLE[total_num] = num;
        total_num = total_num + 1;
        return (total_num-1);
}

int main()
{
        char in_file_name[100];
        printf("Enter the input file name : ");
        scanf("%123s",in_file_name);

        extern FILE *yyin;
        extern FILE *yyout;

        yyin = fopen(in_file_name, "r");

        if(yyin == NULL)
                printf("\nError!!!\nCan't open the file.\n");

        yyout = fopen("output_file.cpp", "w");

        yylex();

        fclose(yyin);
        fclose(yyout);

        return 0;
}
```