**Lab 2**

Quest: Describe the output of a scanner + (Bonus) Use Flex in any way

The following is sample flex output code from my lab1, output.txt which shows the tokens generated from a C file, input.txt in lab1. The code represents the regular language described by the regular expressions in tokenprog.l also embedded in lab1.

**Pre-processor directive : #include <stdio.h>**

This token is as a result of the regular expression

#.\* {printf("\nPre-processor directive : %s",yytext);}

which recognizes <stdio.h> also included in the declaration section for the standard I/O C library. The token is recognized as a pre-processor directive.

**Function : main**

This token is recognized as the main function which acts as an entry point for the program as defined by the expression:

{letter}({letter}|{digit})\*/\(.\*\) {printf("\nFunction : %s",yytext);}

The regular definitions included in the declaration section recognizes the pattern of letters and looks ahead to find () brackets hence recognizes the token as a function.

**Special symbol : (**

This token is recognized as a special symbol as a result of the regular expression

"{"|"}"|"["|"]"|"("|")"|"&" {printf("\nSpecial symbol : %s",yytext);}

**Data type : int**

This token is recognized as the type of data that would be stored in the succeeding variable.

**Identifier : number1**

This token is recognized as a result of the regular expression

{letter}({letter}|{digit})\* {printf("\nIdentifier : %s",yytext);}

that matches the pattern for an identifier. Storage space will thus be allocated for the variable.

**Delimiter: ,**

This token is recognized as it matches the pattern of the regular expression

[,;] {printf("\nDelimiter: %s",yytext);}

that specifies a boundary between separate, independent text.

**Function : printf**

This token is generated as a result of the regular expression

{letter}({letter}|{digit})\*/\(.\*\) {printf("\nFunction : %s",yytext);}

and is the function responsible for writing the string pointed to by format to stdout.

**String : "Enter two integers: "**

This token is generated as a result of the regular expression

\".\*\" {printf("\nString : %s",yytext);}

that matches any number of characters within double quotes.

**Single line comment : // calculating sum**

This token is recognized by the regular expression

"//".\* {printf("\nSingle line comment : %s",yytext);}

which signifies single line comments which are ignored in the latter stages.

**Assignment operator : =**

This token is recognized by the regular expression

"=" {printf("\nAssignment operator : %s",yytext);}

which signifies assignment of a value to a variable.