NED University of Engineering and Technology

Department of Computer Science and Information Technology

Spring 2025 – Third Year Computer Science

CT367 – Theory of Programming Language

ASSIGNMENT # 1

NAME: SHEEZA ASLAM
ROLL NO: CT-22064
SECTION: B
BATCH: 2022

Question 1: Construct a Lex program to count the number of characters, words, constant digits, lines, spaces, tabs, and special characters by making file input have extension .txt.

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

int total_chars = 0, total_words = 0, total_digits = 0, total_lines = 0;
int total_spaces = 0, total_tabs = 0, total_specials = 0;
%}

%%
[a-zA-Z]+      { total_words++; total_chars += yyleng; }
[0-9]+         { total_digits += yyleng; total_chars += yyleng; }
\n             { total_lines++; total_chars++; }
[ ]            { total_spaces++; total_chars++; }
\t             { total_tabs++; total_chars++; }
[^a-zA-Z0-9 \t\n] { total_specials++; total_chars++; }

.              { total_chars++; }
%%

int yywrap() {
   return 1;
}

int main() {
   printf("Enter input (Press Ctrl+Z to stop):\n");
   yylex();

   printf("\nTotal Characters: %d\n", total_chars);
   printf("Total Words: %d\n", total_words);
   printf("Total Digits: %d\n", total_digits);
   printf("Total Lines: %d\n", total_lines);
   printf("Total Spaces: %d\n", total_spaces);
   printf("Total Tabs: %d\n", total_tabs);
   printf("Total Special Characters: %d\n", total_specials);
```

```
    return 0;
}
```

Question 2: Construct a Lex program to identify the email address, DOB and NIC number from the resume file (having extension “.pdf ”).

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

FILE *documentFile;
%}

%%

[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,} { printf("Email Found: %s\n",
yytext); }
[0-9]{2}-[0-9]{2}-[0-9]{4}              { printf("DOB Found: %s\n", yytext); }
[0-9]{5}-[0-9]{7}-[0-9]               { printf("NIC Found: %s\n", yytext); }
.|\n                         { }

%%

int yywrap() {
    return 1;
}

int main(int argumentCount, char **argumentValues) {
    if (argumentCount < 2) {
        printf("Usage: %s <filename>\n", argumentValues[0]);
        return 1;
    }

    documentFile = fopen(argumentValues[1], "r");
    if (!documentFile) {
        printf("Error opening file: %s\n", argumentValues[1]);
        return 1;
    }

    yyin = documentFile;
    yylex();
    fclose(documentFile);

    return 0;
}
```

Question 3: Construct a Lex program to identify (syntax checking) conditional statements from C based source files (having extension ".c").

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

%%

if[ ]*\( { printf("Found: if statement\n"); }
else[ ]+if[ ]*\( { printf("Found: else-if statement\n"); }
else { printf("Found: else statement\n"); }
switch[ ]*\( { printf("Found: switch statement\n"); }
case[ ]+.*: { printf("Found: case statement\n"); }

%%

int main(int totalArgs, char *argList[]) {
    if(totalArgs != 2) {
        printf("Usage: %s <filename.c>\n", argList[0]);
        return 1;
    }

    FILE *sourceFile = fopen(argList[1], "r");
    if (!sourceFile) {
        printf("Error opening file: %s\n", argList[1]);
        return 1;
    }

    yyin = sourceFile;
    yylex();
    fclose(sourceFile);

    return 0;
}
```