Lab Activity 4
ESE 124 (Programming Fundamentals)

Developed by Dr. Alex Doboli
Extended and edited by Mrs. Jenny Chen

Note*: You will earn 30% of your lab grade for completing and submitting your lab work at the end of the lab activity. You must submit the code that you developed for the exercises that you worked on during the lab, and the three associated to-dos. The TODOs should be short. Please don't spend more than 3 minutes on each. You will be graded on your effort during the lab session. You will not be graded on the correctness of the submission.*

The remaining 70% will be earned after submitting your final solutions for the lab exercises, and they will be graded on correctness to the test cases.

For both submissions, each lab exercise is 100 points. The grade for the lab is the average number of points for the submitted exercises.

***The homework exercises** are submitted together with the lab exercises, the latest before the start of your next week's lab session. Each homework exercise is 100 points. The grade for the homework is the average number of points for the submitted exercises.*

Submission: Please follow the instructions posted on Brightspace. Your final solutions are due prior to the start of your next week's lab session.

Lab Activity Description

Goal:

The goal of this lab is to present you the developing of a C program for text scanning.

Scanner programs read text input files and recognize specific items in the text, such as identifiers, numbers in various formats (like integers and decimal numbers), keywords (identifiers with a predefined meaning), operators (+, -, *, / etc.), and special characters, like ; : . . Scanner programs ignore certain characters, like space, new line, and tab. The recognized items are saved in a text output file.

The parts of this lab activity were structured so that at the end you will have a working scanner program.

Lab Exercises

1. Part 1 (1 hour 20 minutes): scanning characters from an input file

Exercise 1: Design a C program that reads characters from a text input file and writes them into a text output file. The input and output files must be identical.

Exercise 2: Create a C program to extract identifiers from text files, outputting them to the result file. An identifier is a sequence of letters, digits, and underscores, always starting with a letter or underscore, allowing both lower and upper-case characters. It ignores the rest of the characters.

Input file:

```
This is _ESE124 Lab_4
Here are some common programming basics:
3 + 1 is not always equal to 3++, nor 3+=1
3 – 1 is not always equal to 3--, nor 3-=1
3 * 5 is not always equal to 3 *= 5
5 / 2 is not always equal to 2.5
2.5 * 2 is not always an integer, unless it is defined as such
% is not a percentage
@ is not an operator, nor a command
^ does not equal exponent.
& is not the same as &&
= is not the same as ==
; ends all lines except loops and conditionals
This is a test case file: you must make your own
```

Output file:

```
This is _ESE124 Lab_4
Here are some common programming basics
is not always equal to nor
is not always equal to nor
is not always equal to
is not always equal to
is not always an integer unless it is defined as such
is not a percentage
is not an operator nor a command
does not equal exponent
is not the same as
is not the same as
```

```
ends all lines except loops and conditionals
This is a test case file you must make your own
```

Exercise 3: Change the C program in exercise 2, so that all identifiers include only upper-case letters. So, if the input file includes the identifier _abCd1_9t the output file will include the corresponding identifier _ABCD1_9T

Output file:

```
THIS IS _ESE124 LAB_4
HERE ARE SOME COMMON PROGRAMMING BASICS
IS NOT ALWAYS EQUAL TO  NOR
IS NOT ALWAYS EQUAL TO  NOR
IS NOT ALWAYS EQUAL TO
IS NOT ALWAYS EQUAL TO
IS NOT ALWAYS AN INTEGER UNLESS IT IS DEFINED AS SUCH
IS NOT A PERCENTAGE
IS NOT AN OPERATOR NOR A COMMAND
DOES NOT EQUAL EXPONENT
IS NOT THE SAME AS
IS NOT THE SAME AS
ENDS ALL LINES EXCEPT LOOPS AND CONDITIONALS
THIS IS A TEST CASE FILE YOU MUST MAKE YOUR OWN
```

Submit the C codes and TODO file for ***Exercises 1, 2 and 3.***

Part 2 (1 hour 20 minutes): scanning operators and special characters

Exercise 4: Design a C programs that recognizes the following operators in the text input file: +, -, *, /, %, @, &, ^. The corresponding text output file will contain only the recognized operators denoted as follows: PLUS for + in the input file, MINUS for – in the input file, MULTIPLY, DIVIDE, MODULO, AT, BITWISE AND, BITWISE OR.

Output:

```
PLUS PLUSPLUS PLUS
MINUSMINUS MINUS MULTIPLY MULTIPLY
DIVIDE
MULTIPLY
MODULO
AT
BITWISE OR
BITWISE AND BITWISE AND BITWISE AND
```

Exercise 5: Extend the previous C program, so that the scanner program also recognizes the operators ++ (INCREMENT), -- (DECREMENT), = (ASSIGN), += (ADD ASSIGN), *= (MULTIPLY ASSIGN), == (EQUAL). The output file will include the words inside the round brackets for each recognized operator.

```
PLUS (INCREMENT) (ADD ASSIGN)
(DECREMENT) MINUSASSIGN
MULTIPLY (MULTIPLY ASSIGN)
DIVIDE
MULTIPLY
MODULO
AT
BITWISE OR
BITWISE AND      BITWISE ANDBITWISE AND
ASSIGN (EQUAL)
```

Exercise 6: Extend the previous C program, so that the scanner also recognizes the following special characters: . (DOT) , (COMMA) ; (SEMICOLON) : (COLON)

Output File:

```
(COLON)
PLUS (INCREMENT) (COMMA) (ADD ASSIGN)
(DECREMENT) (COMMA) MINUSASSIGN
MULTIPLY (MULTIPLY ASSIGN)
DIVIDE (DOT)
(DOT) MULTIPLY (COMMA)
MODULO
AT (COMMA)
BITWISE OR (DOT)
BITWISE AND BITWISE AND BITWISE AND
ASSIGN (EQUAL)
(SEMICOLON)
(COLON)
```

Exercise 7: Integrate all the programs developed in parts 1 and 2 as a single scanner program. This program will be extended next week to complete the scanner program.

Output File:

```
THIS IS _ESE124 LAB_4

HERE ARE SOME COMMON PROGRAMMING BASICS(COLON)

PLUS IS NOT ALWAYS EQUAL TO (INCREMENT)(COMMA) NOR (ADD ASSIGN)
```

IS NOT ALWAYS EQUAL TO (DECREMENT)(COMMA) NOR MINUSASSIGN

MULTIPLY IS NOT ALWAYS EQUAL TO (MULTIPLY ASSIGN)

DIVIDE IS NOT ALWAYS EQUAL TO (DOT)

(DOT) MULTIPLY IS NOT ALWAYS AN INTEGER(COMMA) UNLESS IT IS DEFINED

AS SUCH

MODULO IS NOT A PERCENTAGE

AT IS NOT AN OPERATOR(COMMA) NOR A COMMAND

BITWISE OR DOES NOT EQUAL EXPONENT(DOT)

BITWISE AND IS NOT THE SAME AS BITWISE AND BITWISE AND

ASSIGN IS NOT THE SAME AS (EQUAL)

(SEMICOLON) ENDS ALL LINES EXCEPT LOOPS AND CONDITIONALS

THIS IS A TEST CASE FILE(COLON) YOU MUST MAKE YOUR OWN

Submit C codes and TODO file for ***Exercises 4, 5. 6 and 7.***