COMP500 / ENSE501: Week 7 – Exercise:

EXERCISE NAME: *Running Analysis*

Design (using pseudo code), implement (using C) and test (using Visual Studio Enterprise 2017) a simple program where the user inputs characters to be analysed one character at a time. The user must be prompted for their single character input and be able to repeatedly input another character until they choose to quit. At any time, the user can also choose to enter a "character analysis mode" by inputting the `'#'` symbol.

Declare and define at least three functions in addition to the **`main`** function. Do not use global variables or **`goto`**!

The program must start as follows:

```
Character Running Analysis!
---------------------------

Press '~' to quit at any time...
Press '#' to view current analysis of character input...

Characters input: 0 so far... Your next input?
```

The program must track how many times the user inputs a lower-case character, an upper-case character, and a digit. Any other input is not tracked.

For example, inputting `'n'` increases the lower-case character count by 1, and the overall character input count by 1:

```
Characters input: 0 so far... Your next input? n

Characters input: 1 so far... Your next input?
```

Another example, inputting `'S'` increases the upper-case character count by 1, and the overall character input count by 1:

```
Characters input: 1 so far... Your next input? S

Characters input: 2 so far... Your next input?
```

Another example, inputting `'7'` increases the digit character count by 1, and the overall character input count by 1:

```
Characters input: 2 so far... Your next input? 7

Characters input: 3 so far... Your next input?
```

When the user inputs an invalid character, the program must behave as follows:

```
Characters input: 3 so far... Your next input? @
Invalid input!

Characters input: 3 so far... Your next input?
```

Another example is as follows:

```
Characters input: 3 so far... Your next input? $
Invalid input!

Characters input: 3 so far... Your next input?
```

When the user quits, the following message must be displayed:

```
Characters input: 3 so far... Your next input? ~

In total, 3 valid characters were input! Goodbye!
```

At any time, the user can also choose to enter a "character analysis mode" by inputting the `'#'` symbol.

When entering "character analysis mode", the user must then choose what category of character to view the analysis of: lower-case, upper-case, or digit.

When the user chooses to view the lower-case analysis, the program behaves as follows:

```
Characters input: 11 so far... You next input? #

View current analysis (l/u/d)? l
****************************************
* You have input 4 lower-case characters *
****************************************

Characters input: 11 so far... You next input?
```

When the user chooses to view the upper-case analysis, the program behaves as follows:

```
Characters input: 14 so far... You next input? #

View current analysis (l/u/d)? U
****************************************
* You have input 1 upper-case character *
****************************************

Characters input: 14 so far... Your next input?
```

When the user chooses to view the digits analysis, the program behaves as follows:

```
Characters input: 14 so far... You next input? #

View current analysis (l/u/d)? d
************************************
* You have input 3 digit characters *
************************************

Characters input: 14 so far... Your next input?
```

Ensure each analysis message is surrounded by ***** symbols as shown above.

Ensure each analysis message is plural sensitive, and appropriately includes or excludes the **`'s'`** on the word “characters” depending on the number of characters input!

An example of the completed program is on the following page.

An example of the completed program, with user input:

```
Character Running Analysis
--------------------------

Press '~' to quit at any time...
Press '#' to view current analysis of character input...

Characters input: 0 so far... You next input? n

Characters input: 1 so far... You next input? e

Characters input: 2 so far... You next input? n

Characters input: 3 so far... You next input? n

Characters input: 4 so far... You next input? e

Characters input: 5 so far... You next input? 5

Characters input: 6 so far... You next input? 5

Characters input: 7 so far... You next input? 6

Characters input: 8 so far... You next input? 3

Characters input: 9 so far... You next input? #

View current analysis (l/u/d)? l
****************************************
* You have input 5 lower-case characters *
****************************************

Characters input: 9 so far... You next input? 8

Characters input: 10 so far... You next input? r

Characters input: 11 so far... You next input? P

Characters input: 12 so far... You next input? w

Characters input: 13 so far... You next input? ~

In total, 13 valid characters were input! Goodbye!
```

Follow good programming standards for code layout whitespace, naming and commenting. Ensure your C source code can successfully compile. Test your program with a variety of inputs and ensure the resulting program output is as described above.