

LAB 7 — Control Structures, functions, compile multiple files

Submit the solutions, C programs, using the following command:

```
submit 2031 lab6 lab7b.c lab7b.c
```

The due date is at 3:30 on Wednesday.

1. Problem A

1.1 Specification

Write a C program to count the number of digits ('0', '1', '2', ..., '8', '9') in a line of characters. The program reads from the standard input a line of characters and outputs the number of digits found in the line. Compute the sum of all the digits found and output that sum. If the input line contains no digits, the sum is 0.

1.2 Implementation

The file contains the main function named `lab7a.c`. The main function displays the following prompt before each input line:

```
Enter a line of characters>
```

The given line of characters are captured and will be passed to `num_count` function (which is explained next) as the argument. Note that you need to make the taken line of characters to a null terminated string.

- Make a separate file named `util.c`. In this file make a function with the following signature:

```
void num_count(char[]);
```

The function finds how many numbers are existed in the passed argument and sum of found number and prints them.

- Display on the standard output the number of digits found in the input line and the sum of all the digits, separated by a **tab** character `'\t'`.
- **Hint:** When you find a number represented as character 'x' you can find the integer number by subtracting character '0' from 'x'.

1.3 Sample Inputs/Outputs:

```
indigo 352 % lab7a
```

```
Enter a line of characters>Welcome to CSE2031, "Software Tools".
```

```
4      6
```

```
indigo 353 % lab7a
```

```
Enter a line of characters>0123456789
```

EECS2031
10 45

indigo 354 % lab7a

Enter a line of characters>a b c d e f g h

0 0

indigo 355 %

2. Problem B

2.1 Specification

We want to repeat **Problem A** except this time we consider a sequence of integer numbers. In other words, all sequential numbers are considered as one number. Treat any no-digit character as a separator between numbers. Note that we only considering integer numbers not float point numbers. In other words, '.' is considered a separator.

2.2 Implementation

The file contains the main function named `lab7b.c`. The main function displays the following prompt before each input line:

```
Enter a line of characters>
```

The given line of characters are captured and will be passed to `int_count` function (which is explained next) as the argument. Note that you need to make the taken line of characters to a null terminated string.

- Make a separate file named `util.c`. In this file make a function with the following signature:

```
void int_count(char[]);
```

The function finds how many integers are existed in the given argument and sum of found integers and prints them.

- Make a file `util.h` and use the file for compiling `lab7b.c`.
- Display on the standard output the number of digits found in the input line and the sum of all the digits, separated by a **tab** character `'\t'`.

2.3 Sample Inputs/Outputs:

```
indigo 352 % lab7b
```

```
Enter a line of characters>EECS2031, EECS2021.
```

```
2      4052
```

```
indigo 353 % lab7b
```

```
Enter a line of characters>20 31 20.21.
```

```
4      92
```

```
indigo 354 % lab7b
```

```
Enter a line of characters>20ds 31..3. 20.21.0.0
```

```
7      95
```

Common Notes

All submitted files should contain the following header:

```
/******  
*      EECS2031 - Lab 7  
*      Filename:   Name of file  
*      Author:    Last name, first name  
*      Email:     Your preferred email address  
*      EECS login ID: Your EECS login ID  
*****/
```

In addition, all programs should follow the following guidelines:

- Include the `stdio.h` library in the header of your `.c` files.
- Use `printf` to print text and outputs according to the required formats.
- End each output result with a new line character `'\n'`.
- Do not use any C library function except `getchar()`, `putchar()`, `scanf()`, `printf()`, `fgets()`, `fputs()`, and `sscanf()`.
- **Assume that all inputs are valid (no error checking is required on inputs).**