CIS*1500 – Assignment 3

Due: Sunday Mar 6 – 11:59pm EXTENDED

Write complete C programs for each problem and submit them separately in the dropboxes for Assignment 3A and Assignment 3B respectively. **Make sure you submit the .c file, not the executable.** Your programs must compile successfully with the command:
gcc -Wall -std=c99 your_file.c -lm

(A) [**20 marks**] Given an input file **grades.txt** containing the terms grades for CIS 1500, calculate the class average for each assignment, midterm and final in terms of a percent (i.e., 77%) **rounded** to the nearest percentage point.

The first line of the input file has a single integer specifying the number of students N in the class. The second line contains what each item (there are 7 of them) is out of, and the final N lines are the raw **integer** scores for each student as follows:

ass1_mark ass2_mark ass3_mark ass4_mark ass5_mark midterm final

If each assignment is out of 90, the midterm is out of 50, and the final is out of 100, an input file with 3 students may look like:

```
3
90 90 90 90 90 50 100
81 80 70 75 77 44 77
50 40 60 44 55 33 22
90 88 70 90 44 50 11
```

You may assume that the input file is correctly formatted (you do not need to test for validity of the data). The output to this C program must be written to the file **averages.txt** which for the above input would look like:

```
There are 3 students in this class.
The average for assignment #1 is 82%.
The average for assignment #2 is 77%.
The average for assignment #3 is 74%.
The average for assignment #4 is 77%.
The average for assignment #5 is 65%.
The average for the midterm is 85%.
The average for the final is 37%.
```

(B) [**20 marks**] Given an input file **data.txt** containing random text (letters, numbers, special characters), count the number of occurrences of each digit $0, 1, \ldots, 9$ that appear in the file before the special phrase **END** appears as 3 consecutive characters. If no such substring exists, then just count the number of occurrences in the entire file.

If the input file contains the following data (with no substring **END**):

```
23ED42end;h23E\#\#\$4b2;234324EN99999E
N88
EN
D77
```

then the output should be:

```
Number of 0:  0
Number of 1:  0
Number of 2:  6
Number of 3:  4
Number of 4:  4
Number of 5:  0
Number of 6:  0
Number of 7:  2
Number of 8:  2
Number of 9:  5
```