DATA STRUCTURES

BATCH – B
[THURSDAY FEBRUARY 2, 2017: 2:00 PM – 5:00 PM]

ASSIGNMENTS – 4 CODE: assign04

INSTRUCTIONS: [Total Marks: 25]

i) Read all assignments and each problem has to be answered in the same c file.
ii) Create a .c file following the file name convention: abc-assign04.c
Where abc is your roll number and assign04 is the assignment code
iii) Strictly follow the file name convention and do not use scanf()

PROBLEMS: (There are totally 5 problems)

1) **[Marks: 6 marks]**
Consider the following set of integers:
A = { -8, 2, 70, 11, -36, 29, -67, 41, 13, 3, 6, -5, 35, 52, -96, 24, -17, 29, 37, -41, 53, 47, 61, -53, -71, 14, -11, 7, -82, 57, -12, 42, -29, 97, -202 }

Write a function to sort all negative numbers in increasing order and positive numbers in decreasing order. You can use any sorting algorithm.

Print both sets of numbers in two lines separately.

2) **[Marks: 12 marks]**
Consider the following Paragraph:

A stochastic fractal is built out of probabilities and randomness. It is statistically self-similar. We will look at both deterministic and stochastic techniques for generating fractal patterns. A line is self-similar. A line looks the same at any scale, but it's not a fractal. A fractal is characterized by having a fine structure at small scales, you'll continue to find fluctuations, and cannot be described with Euclidean geometry. If you can say, it's a line, then it's not a fractal. Another fundamental component of fractal geometry is recursion. Fractal has a recursive definition. We'll start with recursion before developing techniques and code examples for building fractal patterns.

Write a function to print a list of terms from the above paragraph. You could use either space or "." or "," as a word delimiter.

A
stochastic
fractal
is
built
out
of
probabilities
and
randomness
It
is
...

Write a function to identify the words that contain the character - '.
Sort these terms in alphabetical order and the print the sorted list of terms.

Write a function to count of the number of terms and sort the terms based on their count. You can use a 2–dimensional array.

Print top 5 terms (decreasing order of their count) from this list.

3) **[Marks: 7 marks]**

Use the following function for random number generator:

```
srand((unsigned int) time (NULL));
```

where `abc` is your roll number. Now you could use `rand()` function to generate unique set of 1000 real numbers in [20, 50]

a) Create a file namely, "`abc-input.txt`" where `abc` is your roll number and write all the above 1000 real numbers to this file.

Output: "`abc-input.txt`" that contains 1000 real numbers in [20, 50]

b) Open another file namely, "`abc-output.txt`" and write the following in this file:

   a. **Write a function** to identify the real numbers between [25.000000, 36.999999] and write them in decreasing order
   b. **Write a function** to identify the numbers that contain at least 3 odd numbers and print the same at the end of the output file.