# INTRODUCTION TO PROGRAMMING – ASSIGNMENT 1

MTech(CS) I year 2020–2021

**Deadline**: 15 January, 2021

Total: 25 marks

**SUBMISSION INSTRUCTIONS**

1. Naming convention for your programs: `cs20xx-assign1-progy.c` (assuming `cs20xx` denotes your roll number and `progy` denotes the program number).
2. To submit the solution files (`.c` or `.h`), ensure that they not password protected and mail them together to `<assignisik@gmail.com>` with the subject line as follows: `MTech (CS) 2020-22 cs20xx Assignment 1`.

**NOTE:** The programs are to be written in C and should be well commented. All programs should take the required inputs from stdin and print the desired outputs to stdout, until otherwise stated.

Q1. The `abundancy` of a natural number $n$ is defined as the rational number $\frac{\sigma(n)}{n}$, the ratio between the sum of divisors of the number and the number itself. A number $n$ is defined as `friendly` if it shares `abundancy` with one or more other numbers. This means there might exist a pair of numbers $i$ and $j$ such that $i \neq j$ but $\frac{\sigma(i)}{j} = \frac{\sigma(j)}{j}$. For example, 6 and 28 are `friendly` with each other because $\frac{\sigma(6)}{6} = \frac{\sigma(28)}{28} = 2$. Let us define a number as `friendly senior` if it shares `abundancy` with at least one number less than it. Write a program to verify whether a number given as user input is `friendly senior` or not. [10 marks]

**Input Format**
The input (to be read from stdin) is a number $n$.

**Output Format**
The output (to be printed to stdout) shows `FRIENDLY SENIOR` if $n$ is a `friendly` number with another number less than $n$, otherwise it shows `NOT FRIENDLY SENIOR`. If $n$ is an invalid input it shows `INVALID`.

**Sample Input 0:**
```
140
```

**Sample Output 0:**
```
FRIENDLY SENIOR
```

**Sample Input 1:**
```
80
```

**Sample Output 1:**
```
NOT FRIENDLY SENIOR
```

**Sample Input 2:**
```
200
```

**Sample Output 2:**
```
FRIENDLY SENIOR
```

Q2. Let us define the `value` of a string as the sum of ASCII values of its characters. For example, `value` of the string "In2Prog" is $641 = (73 + 110 + 50 + 80 + 114 + 111 + 103)$. Write a program that will take a set of strings as inputs and show them in the ascending order of `value` as the output. [10 marks]

**Input Format**
Inputs will be provided as command-line arguments denoting the input strings.

**Output Format**
Output is to be printed on the standard output.

**Command-line Arguments**
`./prog2 <string1> <string2> ...`

**Sample Input 0:**
`./prog2 0c 0b 0a`

**Sample Output 0:**
`0a 0b 0c`

**Sample Input 1:**
`./prog2 20 13 18`

**Sample Output 1:**
`20 13 18`

**Sample Input 2:**
`./prog2 a+b c/d b-c`

**Sample Output 2:**
`a+b b-c c/d`

Q3. Write a program that when executed prints its own source code excluding the comment lines. Note that, the comments in a program could be single line or multi-line. [5 marks]