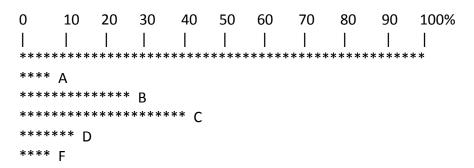## SCSP 1103 Programming Project

Write a C program that graphs a grade distribution (number of A's, B's, C's, D's, and F's) horizontally by printing lines with proportionate numbers of asterisk corresponding to the percentage of grades in each category. Draw the graph such that 50 asterisks correspond to 100 percent (each one corresponds to 2 percent), the horizontal axis goes from 0 to 100 percent with 10 percent in each increment, and each line for data is labeled with letter grade. For example, if there are 1 A, 4 B's, 6 C's, 2 D's, and 1 F, the total number of grades is 14, the percentage of A's is 7, the percentage of B's is 29, the percentage of C's is 43, the percentage of D's is 14, and the percentage of F's is 7. The A row would contain 4 asterisks (7 percent of 50 rounded to nearest integer), the B row 14, the C row 21, the D row 7, and the F row 4, so the graph would look like this:

```
0     10   20   30   40   50   60   70   80   90   100%
|    |    |    |    |    |    |    |    |    |    |
**************************************************
**** A
************* B
******************* C
******* D
**** F
```

Your program should be structured enough. So, you should create several functions to carry out certain tasks such as:
- Reading the number of each letter grade,
- Calculating the percentage of each letter grade as a whole number between 0 and 100 inclusive, and
- Drawing the graph

You are allowed to work in a group of 2 or 3 persons only. All members in a group are expected to participate in doing this project.

Each group is required to:
- Make presentations and demo of their programs in week 14.
- Submit hard copies of flowchart/ pseudo code and the C program at the end of presentation.

Your project will be graded taking into consideration the following aspects:
- quality of design documents (15%) – flowchart/ pseudo code is correct and represent the C program.
- presentation of your C program (5%) – sufficient comments included, good indentation, etc.
- quality of your C program (20%) – program is well structured, use good variable, constant and function names, logic is simple but efficient and correct, etc.
- correctness (50%) – program is compiled successfully, the output is correct and robust (tested for all/ many scenarios), etc.
- creativity and innovative (10%) – user friendly, output is displayed in attractive manner, etc.