ASSIGNMENT 1 – ADA Programming

Your assignment is to implement an Ada program that, among other things, performs a concurrent merge sort.

As described below, you will be creating three files: msort.ads, msort.adb, and main.adb. Please compress these files into a zip file and upload it to your NYU Classes account for this class.

The sections of the Lovelace Ada tutorial that you will find the most relevant are:

- Lesson 2 - Basic Ada Structure (Packages)
- Lesson 13 - Tasks and Protected Types

However, since you will need to declare types, implement procedures, etc., you should be sure to go through at least the following additional lessons in the Lovelace tutorial: 1, 2, 3, 4, 5, 6, 8, 9.

Also, be sure to review the sample Ada programs that I have provided under "Ada Resources" in the course page.

Assignment

Your Ada program will have two parts:

1. A package, Msort, for performing concurrent merge sort.
2. A main procedure, Main.

The Msort Package

You should define a package called Msort whose specification appears in the file msort.ads and whose body appears in the file msort.adb. The Msort package should export (i.e. declare in its specification) the following:

- An integer constant LENGTH, which you should define to be 40. This constant should be used throughout the program, rather than the actual number (so that if you change only this number, your program will still works).
- A type (you can name it what you want) that defines the type of the array to be sorted, namely an array of integers whose values range between -300 and 300, inclusive. The size of the array should be LENGTH.
- A procedure Sort(A), where the parameter A is an array of the type you defined above. After the call to Sort, the elements of A should be in sorted, increasing order.

Within the body of the Msort package, you will define the Sort procedure. This procedure should sort the array parameter A using a concurrent merge sort algorithm. You can use any merge sort algorithm you like (a description of several different merge sort algorithms is discussed in the Wikipedia page on merge sort). However, at any point in the program where sorting occurs on two or more portions of the array, the sorting of those portions must occur concurrently (the same Wikipedia page also discusses parallel merge sort which indicates one way to do this).

The Main Procedure

In a separate file, main.adb, you should define the procedure Main. It should declare an array of the type defined in the Sort package, read in values for the elements of the array, call Sort to sort the array, and also compute the sum of the elements of the array. It should accomplish this using at least three tasks, as follows:

- Reader: This task should read in LENGTH integers from the terminal and write them into the array.
- Sum: This task should compute the sum of the elements of the array.
- Printer: This task should print the elements of the array and then print the sum computed by the Sum task.

The actual call to the Sort procedure should be from the body of the Main procedure, not from within one of the above tasks. Of course, the Sort procedure shouldn't start running until all the numbers have been read into the array, and the printing of the array shouldn't happen until after the sorting is complete. Furthermore, the computing of the sum and the printing of the elements of the array after it has been sorted should happen concurrently. That is, the running of the above tasks should happen as concurrently as possible.

In order to avoid having to type in 40 numbers every time you run your program, simply create a text file containing the 40 numbers, separated by spaces. Then, when you run your program, redirect that file to the standard input by typing

./main < input.txt

assuming that the input file is called input.txt. Since your program will be reading from standard input, the Ada Get procedure can be called, without worrying about files.

Let me know if you have any questions. If you get stuck, feel free to email me.