Project 1 CS 310 Kutztown

Points: 25
Due: TBD, via the turnin script. No late submissions accepted
Purpose: Intro to Ada

Assignment: In this assignment, you will write a program to acclimate yourself to the Ada programming language. Write a program that reads up to N integers into an array. It then sorts the data and presents the user a menu, as in the figure. The user makes their choice and the program responds accordingly, looping until the user uses the Q option to exit the program.

```
Select:
P)rint Array
L)ook for Element
C)hange an Element
I)nsert New Element
F)ind Average
Q)uit
```

Your program will implement each option using one or more subprograms. Your main routine (no longer named *main*!) may contain nothing more than function calls, a loop, and a multiple alternative (Ada equivalent to the C++ switch).

You must have a function for each of the following tasks:

- Print the array, well-formatted, with each value below its index (starting from index 1)
- Sort the array
- Search the array
    - Input an element from the user, search the array for the element, and return whether the search succeeded.
- Alter an element of the array
- Determine whether the array is full
- Insert a new element, if there is room.
- Determine whether an element is in proper order, relative to its immediate neighbors
- Determine the exact average of all values in the array

Notes:

- There are several Ada resources on the course web page and of course all over the internet.
- A new element can be inserted at the end of the array, in the index after the last element, if that index exists. If there isn't room, this function should return a value to indicate this. Following insertion, the list is to be sorted, **only if necessary**. Don't forget to update the variable holding the actual number of values in use in the array
- Declare constants for the boundaries of the integer array. When you turn in your project, make the constants equal to 1 and 10.
- Declare an array type for the array of integers.
- The search function must be a binary search, since the array is sorted. This search returns the index where found, 0 if not found.
- The alteration function is implemented in the *demo1.adb* example. You still need subprograms to input the index and value.
- To determine if an element is in proper order, there are three cases: Element is first in array, element is last in array, or element is in interior. For the first two cases, you need only compare with one element. In the latter, you must determine if it is between its left and right neighbors. This function will be called when the value of an element changes or is added.
- To determine whether an index is first or last in the list, you are to use a tick mark operation. All for loops must use at least one tick mark operation, as in the *demo1* example.
- The average must be a real number. You will need to employ type casting (coercion) to accomplish this. The summing function is used as part of this computation.
- You may use any sort method for your sorting procedure. You must write a separate swap, properly implemented.
- You must properly declare parameter types **in, out** and **in out**, and declare functions and procedures appropriately, according to their intended tasks.
- The array is to be re-sorted if an element is changed or added, IF that element is out of order (use the function for this).
- Input is case insensitive.
- You must employ a consistent style. Since C++ and Ada conventions differ, you are free to pick and choose, BUT once chosen, style must be consistent. For example, if you start off capitalizing the first letter of variable names, all variable names must follow that style.
- The use of global variables is prohibited, without justification. **Pass data to/from subprograms properly, via parameters.** To enforce this, place global variables right above the main program rather than on top.
- Your program MUST be properly documented. Use a proper id block and a header on each function. Provide proper attribution for supplied routines. This is a junior/senior course. Severe penalties will be assessed for inadequate documentation.
- Set up turnin now, as it is the only method of submission of projects. Not having it set up is no excuse for late submission.

Turnin:

Turn in your project using the turnin script. Your file is to be named **p1.adb** (all lower case). There will be a 2 point penalty for non-compliance