COE 205, Term 091

Computer Organization & Assembly Programming

Programming Assignment# 3
Due date: Saturday, Jan. 2, 2010

Q.1. Gnome sort is a sorting algorithm which is similar to insertion sort, except that moving an element to its proper place is accomplished by a series of swaps, as in bubble sort. The name comes from the supposed behavior of the Dutch garden Gnome in sorting a line of flowerpots. The algorithm always finds the first place where two adjacent elements are in the wrong order, and swaps them. It takes advantage of the fact that performing a swap can introduce a new out-of-order adjacent pair only right before or after the two swapped elements. It does not assume that elements forward of the current position are sorted, so it only needs to check the position directly before the swapped elements.

The pseudo code for the Gnome sort algorithm is given below:

```
GnomeSort(address array A, size of array A){
  i := 1
  j := 2
  while i < size
    if A[i-1] <= A[i]
         i := j
         j := j + 1
     else
         swap A[i-1] and A[i]
         i := i - 1
         if i = 0
            i := j
            j := j + 1
}
```

(i) Write a procedure, GnomeSort, to implement the Gnome sort algorithm. All input parameters are to be passed on the stack.

(ii) Ask the user to enter the number of integers to be sorted, n.

(iii) Ask the user to enter an array of n integers and read it.

(iv) Use the GnomeSort procedure you implemented to sort the array, IntArray.

(v) Display the array, IntArray, after sorting.

A sample execution of the program is shown below:

Enter the number of integers to be sorted: 5
Enter an array of 5 integers:
2 1 3 5 6
Array after sorting is:
1 2 3 5 6

The solution should be well organized and your program should be well documented. Submit a soft copy of your solution in a zip file. Your solution should be submitted in a word file that contains the following items:

- *i) Your name and ID*
- *ii) Assignment number*
- *iii) Problem statement*
- *iv) Your solution along with the code*
- *v) Discussion of what worked and what did not work in your program. Include snapshots that demonstrate the working parts of your program. If things did not work and you attempted to solve them, mention that and write about the difficulty that you have faced.*

The soft copy should also contain both source code file (i.e. .asm) and the executable file (i.e. .exe).