

COE 205, Term 082
Computer Organization & Assembly Programming

Programming Assignment# 3
Due date: Saturday, May 30, 2009

Q.1. Insertion sort is a simple sorting algorithm in which the sorted array (or list) is built one entry at a time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort. However, insertion sort provides several advantages:

- simple implementation
- efficient for (quite) small data sets
- efficient for data sets that are already substantially sorted
- more efficient in practice than most other simple quadratic (i.e., $O(n^2)$) algorithms such as selection sort or bubble sort: the average running time is $n^2/4$, and the running time is linear in the best case

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

```
InsertionSort (address array A, length array A)
begin
  for i := 1 to length[A]-1 do
  begin
    value := A[i];
    j := i-1;
    while j ≥ 0 and A[j] > value do
    begin
      A[j + 1] := A[j];
      j := j-1;
    end;
    A[j+1] := value;
  end;
end;
```

- (i) Write a procedure, InsertionSort, to implement the insertion 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 InsertionSort 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).