

## CSIT-816 Introduction to Computer Architecture and Organization

**Project #1. Due: December 7, 2004.**

**LATE ASSIGNMENTS (PROJECTS, HOMEWORK, ETC.) WILL NOT BE ACCEPTED!!!**

**YOU WILL GET "0" POINTS FOR NOT SUBMITTING YOUR WORK ON TIME. COMPLETE OR INCOMPLETE ASSIGNMENTS WILL BE ACCEPTED AND GRADED ONLY IF SUBMITTED ON TIME.**

The objective of this project is to implement three sorting algorithms using Microsoft Assembly tools (MASM 6.15).

Your program should be able to handle any array size up to 256. Your program should be interactive. First it will ask the user which sorting algorithm to use (a choice of three). See below for algorithm descriptions. Then the program will ask the user to enter the array size. The array size should be from 1 to 256. Lastly, ask the user to enter all integers. After the array is sorted, your program should display the entire array. Here is a sample output from the program:

```
A[0] = 10
A[1] = 17
A[2] = 24
A[3] = 38
A[4] = 50
A[5] = 54
A[6] = 55
A[7] = 89
A[8] = 92
A[9] = 99
```

Here is the C++ code for three sorting algorithms:

```
void insertion_sort(int A[], int n) {
    for (int i = 1; i < n; i++) {
        for (int j = i; (j > 0) && (A[j] < A[j - 1]); j--)
        ) {
            int temp = A[j];
            A[j] = A[j - 1];
            A[j - 1] = temp;
        } // for j
    } // for i
} // insertion_sort
```

```
void bubble_sort(int A[], int n) {
    for (int i = 0; i < n-1; i++) {
```

## CSIT-816 Introduction to Computer Architecture and Organization

```
        for (int j = n - 1; j > i; j--) {
            if (A[j] < A[j - 1]) {
                int temp = A[j];
                A[j] = A[j - 1];
                A[j - 1] = temp;
            } // if
        } // for j
    } // for i
} // bubble_sort
```

```
void selection_sort(int A[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int low_index = i;
        for (int j = n - 1; j > i; j--) {
            if (A[j] < A[low_index]) {
                low_index = j;
            } // if
        } // for j
        if (low_index != i) {
            int temp = A[i];
            A[i] = A[low_index];
            A[low_index] = temp;
        } // if
    } // for i
} // selection_sort
```

For an extra bonus points give the user a choice to read the input from a text file and write the output into a text file. Ask the user to specify the input and output file names.