50 pts

Name: _____

Class Day / Time: _____

Due Date: _____

Assignment #7 – Assembly - Searching and Sorting

In this assignment you will create and implement procedures for performing a sequential search, a binary search and a bubble sort for an array of bytes. You can use the assembly code presented in the class. You will need to create the following assembly procedures:

- a procedure that sorts an array of bytes using a bubble sort.
- a procedure that searches an array of bytes using a sequential search and returns in EAX the appropriate index in the array of bytes where the key was found.
- a procedure that searches an array of bytes using a binary search and returns in EAX the appropriate index in the array of bytes where the key was found.
- a procedure that outputs an array of bytes

The main program will perform the following steps:

1) Call the procedure to **output** the array of bytes
2) You will **input integer** from the console and store them in memory. Label the memory locations for the integers as *searchNum*
3) Use *searchNum* to call the procedure that performs a **sequential search** in the array of bytes. Output the index # that represents where the item was found
4) Repeat item 2 and 3 four times
5) Call the procedure that performs the **bubble sort** in the array of bytes
6) Call the procedure to **output** the array of bytes
7) You will **input integer** from the console and store them in memory. Label the memory locations for the integers as *searchNum*
8) Use *searchNum* to call the procedure that performs a **binary search** in the array of bytes. Output the index # that represents where the item was found
9) Repeat item 7 and 8 four times

You will use the following array of bytes definition:

```
.data
 array BYTE 4, 1, 7, 12, 8, 13, 9, 21
```

Draw a flowchart and **write the corresponding assembly language program** in the flowchart for the program you are implementing. It is not necessary to indicate memory locations in the flowchart, but include **all the variable** declaration as part of the flow chart.

Implement the program; test the program a number of times with different input numbers. You will need to **turn in** a test run for the program as shown below.

```
Index #0:  4
Index #1:  1
Index #2:  7
Index #3:  12
Index #4:  8
Index #5:  13
Index #6:  9
Index #7:  21

Enter an integer to search for:  9
The integer 9 was found in index #6.

Enter an integer to search for:  6
6 was not found!

Enter an integer to search for:  21
The integer 21 was found in index #7.

Enter an integer to search for:  4
The integer 4 was found in index #0.


Performing Insertion Sort!

Index #0:  1
Index #1:  4
Index #2:  7
Index #3:  8
Index #4:  9
Index #5:  12
Index #6:  13
Index #7:  21

Enter an integer to search for:  12
The integer 12 was found in index #5.

Enter an integer to search for:  21
The integer 1 was found in index #7.

Enter an integer to search for:  2
2 was not Found!

Enter an integer to search for:  1
The integer 1 was found in index #0.
```

Turn in (STAPLED IN THIS ORDER)

1. The **FIRST PAGE** of this assignment as a coversheet
2. Include the **flowchart** properly documented. The listing of **.asm source code** properly documented.
3. The output from the program, either pasted into .asm source code or using print screen.