**CS47: Section 1, Fall 2020**

**Program #2: Sorter (100 Points)**

**Due Date: Oct 27, 2020**

Description:

You are responsible to write an assembly function to perform sorting of a list of integer key values. This function can sort either in ascending or descending order. You will use the bubble sort algorithm for this function. This function will use the C function signature but everything within this function should be assembly code using the ASM block similar to the assembly example shown in class.

Program Specification:

int sorter ( long\* list, long count, long opcode );

list – the starting address of the list of integers to be sorted

count – total number of integers in the list

opcode -

1. Ascending

2. Descending

Additional Information:

* You must use the XCHG instruction to swap your array element values
* The input list stored in a text file with the filename being passed by command line argument.
* The output is display on the screen.
* You will first sort the list based on command line parameter in either ascending or descending order and then print it out.
* You must add comments to each line of your ASM code inside sorter().
* To get full credit, you need to use the same block of code for ASC and DESC. If you use two separate blocks of code, then your maximum score will be <=80%.

Program Checklist:

For program #2, you will submit your p2.cpp and a file named readme.p2 text file with any comments (things that work or not work about your code besides the required header (name, student ID, OS type, 32/64 bit, and compiler info). You MUST be able to compile and run the program from the command line). If we can't test your program, you're not going to get much credit for your work!!! The output of your program should match exactly the output of my example (p2test.out) using the same input files. Here is the bubble sort algorithm which you need to implement in assembly.

void sorter (long\* list, long count, long opcode)

{

long x, y, temp;

for (x = 0; x < count - 1; x++)

for (y = x; y < count; y++)

if (list[x] > list[y])

{

temp = list[x];

list[x] = list[y];

list[y] = temp;

}

}