

# CSE 378: Machine Organization and Assembly Language

## Assignment #2

*(Assigned October 6, 2000, Due October 13, 2000 by 5:00)*

### 1 Bubble Sort

For the first part of this assignment you will convert a C implementation of bubble sort to MIPS assembly. The listing for bubble sort is given below.

```
void bubble_sort( int* array, int array_size ) {
    for( int pass=1;pass<array_size;pass++ ) {
        for( int i=0;i<array_size-pass;i++ ) {
            if( array[i] > array[i+1] ) {
                int temp = array[i];
                array[i] = array[i+1];
                array[i+1] = temp;
            }
        }
    }
}
```

To aid you in the process of converting this subroutine, two assembly files have been provided: 'main1.s' and 'hw2.1.s' (you can find them on the course web page). The first contains a main function that calls your bubble sort function and then prints the resulting array to the console. You won't need to make any changes to this file unless you want to try different arrays (in which case you can change the array definition in the .data section and the array size in the procedure call). The second file contains a simple skeleton of a bubble sort procedure that you will need to fill in with the actual code. You should ONLY turn in the second file. Using two files is simple in X-Spim, just load them both separately and press run. You can also copy the contents of the second file to the bottom of the first; just remember to put the bubble sort procedure back into its own file when you are done. Instructions for using the electronic turn in program will be sent out to the mailing list later this week and will also be covered in next Thursday's sections.

### 2 Palindrome

For the second part of the assignment you will write a subroutine that takes a string and returns a one if it is a palindrome and a zero otherwise. A palindrome is a word like "radar" that reads the same both forwards and backwards. To return a value in assembly simple set the register v0 to either 0 or 1 at the end of your subroutine. As in part one, a skeleton main (main2.s) and a skeleton procedure (hw2.2.s) have been provided. You should only turn in hw2.2.s and NOT main2.s.