

COE 205, Term 101

Computer Organization & Assembly Programming

Programming Assignment# 3

Due date: Saturday, Dec. 11, 2010

- Q.1.** Cocktail sort is a variation of bubble sort and sorts in both directions on each pass through the list. The first rightward pass will shift the largest element to its correct place at the end, and the following leftward pass will shift the smallest element to its correct place at the beginning. The second complete pass will shift the second largest and second smallest elements to their correct places, and so on. After i passes, the first i and the last i elements in the list are in their correct positions, and do not need to be checked. By shortening the part of the list that is sorted each time, the number of operations can be halved.

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

```
procedure cocktailSort( A : list of sortable items ) defined as:
  // `begin` and `end` mark the first and last index to check
  begin := -1
  end := length( A ) - 2
  do
    swapped := false
    // increases `begin` because the elements before `begin` are in
correct order
    begin := begin + 1
    for each i in begin to end do:
      if A[ i ] > A[ i + 1 ] then
        swap( A[ i ], A[ i + 1 ] )
        swapped := true
      end if
    end for
    if swapped = false then
      break do-while loop
    end if
    swapped := false
    // decreases `end` because the elements after `end` are in correct
order
    end := end - 1
    for each i in end to begin do:
      if A[ i ] > A[ i + 1 ] then
        swap( A[ i ], A[ i + 1 ] )
        swapped := true
      end if
    end for
  while swapped
end procedure
```

- (i) Write a procedure, **CocktailSort**, to implement the Cocktail 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 **CocktailSort** 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).