Assignment 2
Assembly Language Programming

due
Friday, February 5 1999
8:30 AM

Question 1

Write an 80x86 assembly-language program that:

- reads exactly 10 characters from the user and stores them in an array

- sorts the ten characters into ascending order by ASCII code using a bubble sort

- prints the list of sorted characters on the next line

For example, if you ran your program and then entered the string `a2jHa8kk=N` your program would start a new line after you entered the `N`, print `28=HNaajkk`, and return to DOS.

Your code should make use of only these DOS interrupts: int 21H function 1 (AH=1) to read a character, int 21H function 1 (AH=2) to print a character, and int 20H to return control to DOS.

A bubble sort repeatedly loops over the pairs of values stored in the array and swaps the two values if they are out of order. This process is repeated until no swaps take place.

Comments are particularly important when writing in assembly language. The comments in your code must include at least: a file header with the file name, the purpose of the code, your name, student number, and date; a function header for each function describing the purpose of the function, the inputs, outputs, and registers altered by each function (if any); and a short in-line comment for each part of each function explaining the purpose of that section of code. The in-line comments should be a high-level explanation of your algorithm similar to pseudo-code rather than an explanation of the effects of each instruction. For example, use:

```
xor     ax,ax       ; reset count to zero
mov     count,ax
```

Do *NOT* (!) write comments such as this:

```
xor     ax,ax       ; exclusive-or AX with AX
mov     count,ax     ; store AX in location count
```

or even

```
xor     ax,ax       ; set AX to zero
mov     count,ax     ; store AX in count
```

Assemble and test your code using the free valarrow DOS assembler and linker available from the course web page.

Submit a listing of your code and a print-out of its output. You can redirect the output of your program to a file using '>' (e.g. `asg2 >`*filename*). Submitting output that was not generated by the program in the accompanying listing will be considered academic dishonesty and will result in a mark of zero or worse.

Bonus marks will be awarded for the shortest and for the easiest-to-read solutions.

You may find it useful to first write and test your program in a high-level language (e.g. C) and then convert it to assembly language. To start a new line you must print carriage return (0DH) and line feed (0AH) characters.