COE 205, Term 062

Computer Organization & Assembly Programming

Programming Assignment# 2
Due date: Saturday, May 5, 2006

Q.1. You are required to write a procedure, **CharDist** that receives as arguments the address of a string of characters and its length passed through the stack. The procedure will count the number of times alphabetic characters ('a' to 'z' both upper and lower case) occur. Then, it displays a histogram chart showing characters distribution by displaying N stars for a character that is encountered N times. The procedure should preserve all used registers. Write a program to call this procedure to display character distribution of any string entered by the user. A sample execution of the program is given below.

Enter a string of characters: I like assignment#2!!

Character Distribution:

```
A  *
E  * *
G *
I  * * *
K *
L *
M *
N * *
S * *
T *
```

Q.2. Write a procedure, **SelectionSort**, to sort an array of integers (i.e. 32-bit signed numbers) in an **ascending** order. The number of integers to be sorted and the address of the array to be sorted are assumed to be passed on the stack. The procedure should maintain the content of all registers to their state before its execution.

The pseudocode for the **SelectionSort** procedure is given below:

```
SelectionSort (Array, Size)
    for (position= 0 to Size-2)
        MinValue = Array[position]
        MinPosition = position
        for (j=position+1  to Size-1)
            if (Array[j] < MinValue) then
                MinValue = Array[j]
                MinPosition = j
            end if
        end for
        if (position $\neq$ MinPosition) then
            Array[MinPosition] = Array[Position]
            Array[Position] = MinValue
        end if
    end for
end SelectionSort
```

Write a complete program, showing the place of procedure definition, to use the procedure **SelectionSort** to sort the Array given below:

```
Array DD 10, 2, 0, 15, 25, 30, 7, 22
```

Note that the Content of Array after sorting will be:

```
Array DD 0, 2, 7, 10, 15, 22, 25, 30
```

The solution should be well organized and your program should be well documented. Submit a soft copy of your solution in a zip file. The soft copy should include a Readme file indicating the file names containing the solution and whether it works or not. The Readme file should also contain your name and ID. Submit both source code file (i.e. .asm) and the executable file (i.e. .exe).