CIS240 FINAL ADD 2 NUMBERS CALCULATION PROJECT

Introduction:

You will be creating a simple Calculator, all it needs to do is add two decimal numbers of a fixed length, and generate an output for all possible values of the two number inputs.

The calculator must first accept the ASCII keyboard information then convert that information into Binary Coded Decimal (BCD) so that the CPU can calculate a result (just subtract 30h from each number). Also the once the calculation is complete the BCD value must be converted back (add 30h) and sent to the display monitor.

To get full credit (50 points):
The inputs must be able to accept either no numbers (and interpret them as a zero) or two numbers of the SAME lengths at least 5 digits (this insures you will have to use arrays and loops).

The output must only display the digits that are part of the output number with no leading zeros. Therefore, the output will be from 1 to N digits depending on the input values to be added.

Use subroutines for the conversions from ASCii to BCD and BCD to ASCii or any other repetitive set of code

Display the final carry of any two numbers added

For extra credit (maximum amount of extra credit possible is 25 points depending on your implementation of one or more of the following):

- Add two numbers of Different Lengths; therefore, you need to process the proper input length of data for each number(10points).
- Add two numbers with decimal points included in the number entry at any location is worth more(15points)
- Some added feature you discuss with the instructor

You must turn in:
A print out of your code, any flow charts or pseudo code you created, if you use a different algorithm than provided submit that, and a paragraph explaining your debugging method and some major problems you encountered and how you solved them. You must demonstrate your code, and be ready to answer any question regarding your code operation from the work you submit.

You must all do your own work, it is ok to consult each other and brainstorm but any work that is copied will result in a shared score between participants.

On the next sheet is a BCD algorithm in pseudo code that you can use as an outline to design your assembly code.

The project is due up to the day of the final.

For two number A and B of length n and m there is a Sum result C of length l

A_n A_{n-1} ….A_1 A_0
B_m B_{m-1}… B_1 B_0

C_l C_{l-1}……C_1 C_0

Pseudo Code:

Zero any array memory

Input A
Input B

As the digits are being input convert the them from ASCII to Decimal (subtract 30h), and store them in two arrays

Find the length of A as n and B as m

if n > m then l = n
if m > n then l = m

```
; add the two numbers one digit at a time and take into account the carry
For i = 0 to l-1
        t = A(i)+B(i)+C(i)              ;add two digits plus previous carry
        If t >= 10 Then
                C(i)=t-10
                C(i+1) = 1              ;store carry into next place
        Else
                C(i)=t
        End If
Next
```

Convert the output array to ASCii (Add 30h to each digit)

Output C to monitor (include last carry if there is one)

In order to use the above algorithm you must reverse the order of the input arrays, then reverse them again in order to print them out in the correct order. Otherwise, you will have to down count your array loop by starting at the array end which contains the LSD down to the beginning which contains the MSD.

If you attempt to add numbers of different lengths to get extra credit then reversing the order before adding will make that process much easier(in fact that is all you have to do); however if you reverse the order of the input array to make adding easier then you have to undo the reversal to print them out correctly.