CIS 21JA - Assignment 4

There are 2 parts to Assignment 4.

**Part 1**  
Download assignment4.asm and use it to write a program that calculates the total amount of money from a given number of dollar bills and coins.

These are the steps for the program:

* Ask the user for the number of ten dollar bills, five dollar bills, one dollar bills, quarters, dimes, nickels and pennies (there should be 7 separate prompts, in that order). Read in one numeric value at a time.  
  You can assume the user will enter a range between -10 and 10, inclusive.
* The number of bills and coins can be positive or negative: positive means the user is receiving money, negative means the user is giving away money.  
  - Example of positive input  
   Someone gives you 2 quarters and 3 dimes: you enter 2 and 3 at the appropriate prompts.  
  - Example of negative input  
   You pay someone 8 dollars: you enter -8 at the prompt for one dollar bills.  
  - Example of both positive and negative input

At a cash register you pay 3 dollar bills and 3 pennies and get back 2 quarters: you enter -3 at the dollar prompt, 2 at the quarter prompt, and -3 and the penny prompt. The program will calculate the total of -2 dollars and -53 cents, which means you paid out $2.53.

* After reading in all input, calculate the total amount of money from the number of bills and coins entered.
* Print the total amount of money as a separate number of dollars and number of cents, along with a text explanation. The output should be on one line of text. See sample output:

Enter number of ten dollar bills: -8

Enter number of five dollar bills: -5

Enter number of one dollar bills: 9

Enter number of quarters: 2

Enter number of dimes: 4

Enter number of nickels: 5

Enter number of pennies: 10

The total is -94 dollars and -75 cents

Press any key to continue . . .

Additional requirements to get everyone to think about data size and registers:  
A. The range of user input is -10 to +10, which means the range of the amount of money is +/- 164.10 dollars. Use the *smallest* register size that can handle this range of data. This means the calculation should not be using DWORD size registers. But keep in mind that IO procedures to print and read data need to use 32-bit registers.  
  
B. Use no memory variable to store the results of the calculation. Memory variables are only for text strings. Use registers instead, and as a review of module 3, these are the available registers: EAX, EBX, ECX, EDX, EBP, ESI, EDI

**Part 2**Follow the instructions in assignment4.asm to answer the questions on arithmetic flags.