CIS 21JA – Lab 4

There are 2 parts for this lab. Make sure you complete both parts.

Part 1 (10 pts total)

Go to the section labeled “Part 1” in lab4.asm and add code to the existing program to calculate the number of coins in a cash payment.

Example: a customer buys an item for $4.35 and gives a $5 bill for payment.

The change amount would be 65 cents, and the program will print out that the change is:

2 quarters, 1 dimes, 1 nickels, 0 pennies  
(It’s okay to end all the coin names with ‘s’ even if there’s 1 coin – don’t tell your English instructors)

Here are the steps to complete the program (6/10 pts):

1. Create a constant called MAX and set it to 99 (this is the max change amount of 99 cents)
2. The change amount is simulated by a random number between 0 and MAX.  
   Call the randomRange procedure to get a random number between 0 and MAX (make sure you use MAX and not 99 in the code).   
   Run the sample code in lab4.asm to see how to call randomRange, then modify the code so that randomRange returns a random number between 0 and MAX. This random number will be the change amount.
3. From the change amount, calculate the number of quarters, dimes, nickels, and pennies.  
   Here’s the pseudocode for converting the change amount to coins:  
    Pseudocode Example

|  |  |
| --- | --- |
| num quarters = change amount / 25 update change amount to remainder | 65 / 25 = 2 (quarters), remainder: 15 |
| num dimes = change amount / 10  update change amount to remainder | 15 / 10 = 1 (dime), remainder: 5 |
| num nickels = change amount / 5  num pennies = remainder | 5 / 5 = 1 (nickel), remainder: 0 (pennies) |

1. Print the result in this format: N quarters, N dimes, N nickels, N pennies  
   where N is the calculated number of each coin.  
   Make sure the output string ends with a ‘\n’ (in assembly) so that it’s on a line by itself.

Sample output:

Change is 36 cents

1 quarters, 1 dimes, 0 nickels, 1 pennies

DON'T MISS these additional requirements:

1. (2/10 pts) Except for text string variables, which are arrays, the program should use NO memory variables to store numbers. What should you use instead, which are faster to access than variables?

2. (2/10 pts) Given the range of data in the program, use the \*smallest\* data size (not DWORD) in all your calculation.   
 But make sure you use DWORD data when printing numbers because the Irvine32 IO library works with 32 bit data.

This is an exercise with 3 learning objectives:

- to practice using different size data

- to be aware of when data can change size in an arithmetic operation

- to be aware of when you should change the size of the data

Part 2 (5pts total)

Follow the instructions in Part 2 of lab4.asm to fill in the flag values and your explanation.