

## ARM Lab #2

Your program will simulate the one-day operation of a simple all time teller machine. The all time teller will dispense, upon request, \$20, and \$10 bills up to \$200 total per customer. Money amounts that can not be made up of \$20 and \$10 bills will result in an error message (for example, \$25 is not supported; neither is \$51. However, \$30, \$50, etc. are supported). Your all time teller should use \$20 bills where possible. For example, \$50 should be made up of two \$20 bills and one \$10 bill, rather than, for example, one \$20 bill and three \$10 bill. Your machine will be loaded, in the morning, with 50 \$20 bills and 50 \$10 bills. Your machine will support exactly 10 customers, no more and no less, (in the single day of operation), or up to the total amount of money within the machine (\$1500 dollars =  $50 \times \$20 + 50 \times \$10$ ). Ten customers legally requesting \$200 each would be \$2000 total, so it is possible to run out of money on a given day. When a customer requests more money than that remaining in the all time teller machine, then the teller should give an error message to the effect that not enough money is available. If that customer then re-tried with a smaller amount of money, that customer, for this purposes of this program, would be considered to be another customer (in other words, the customer would count more than once in the count of ten customers).

When your all time teller machine runs out of \$20 bills, then it will use \$10 to make up all amounts until it runs out of those also. If your all time teller machine, for some reason, should run out of \$10 bills before it runs out of \$20 bills (I don't really believe that's possible, given the restriction that \$20 bills should be used first, and the small number of customers, but just in case someone should be worried about it), the all time teller should dispense \$20 bills until it runs out of those. In such a case, it would be able to accept only amounts that are multiples of \$20 (and would have to give appropriate error messages).

Your program should give the number of \$20 bills dispensed, and the number of \$10 bills dispensed to make up the total.

Hints:

Have one variable that is a downcount of \$20 bills, initialized to 50. Have one variable that is a downcount of \$10, initialized to 50. Count down these variables as money is dispensed.

Have two variables that serve as upcounters of \$20 bills and \$10 bills for a particular transaction.