

Teller Machine

Objective: The objective of this assignment is to have the students build upon the skills from ARM Lab 4 for a more complex ARM Assembly program. The students will have to track multiple items forcing the use of memory locations or the stack to save and restore data values.

Your program will simulate the one-day operation of a simple teller machine. The teller will dispense, upon request, \$20, and \$10 bills up to \$200 total per customer. Your software will perform the following:

1. Display a welcome message and instructions on how to make a valid withdraw.
2. Reject any invalid withdraw requests. Examples: must be a multiple of 10, and must not be greater than \$200. Assume the customer can only enter numbers on a 10-digit keypad (0-9).
3. When filling a valid withdraw request disperse \$20 bills first.
4. When all the \$20 bills are gone disperse \$10 bills to fulfill the request.
5. Limit the total number of withdraws to 10. Invalid requests do not count as a transaction.
6. If there are insufficient funds on hand for a withdraw inform the customer and ask them to enter a lower request.
7. Initial inventory of bills is 50 of each type (50 - \$20 bills and 50 - \$10 bills). Once the inventory of bills is depleted or maximum withdraws is reached display an ending message and stop the program.
8. When the program ends print out: Total number of valid transactions, Number of \$20s distributed, Number of \$10s distributed, total amount of money distributed, remaining funds on hand.
9. The program will allow the entry of a secret code that when entered will display the following data:
 - a. Inventory of \$20 and \$10 bills
 - b. Remaining balance on hand
 - c. Current number of transactions
 - d. Total distributions made so far.

Suggest the secret code be when a “-9” is entered for the withdraw amount request.

Prework:

Students need to have a file which, at the very least contain the following:

1. Proper header information that include commands that assemble, link, run the file and run the file using the debugger that is specific for their file.
2. Data section that defines all the output strings.
3. Code comment sections that describe the major sections of the code with the output loop clearly identified.

CS309 Lab 5 ARM Advance Program

Teller Machine

If the student already has a working program this prework check is not required.

This lab was assigned

Fall 2019, Fall 2018, Spring 2017

Teller Machine

Student _____

Lab Instructor: _____

Total Score: _____

Late Penalty? No _____

Yes: _____

Feature	Points off
Code Comments/Documentation <ul style="list-style-type: none"> - Class, Term, Author, Date. - Purpose of software - Documentation for the start of each loop. - Documentation for error checking - In-line comments that explain why the following code exists. 	
Welcome message and Teller Machine instructions displayed and are clear.	
Limit of 10 transaction – Withdraw small amounts until 10 transactions are reached. Switch between multiple of 20 and not multiples of 20 to ensure right number of \$20s and \$10 are dispersed.	
Rejects invalid withdraw (Negative, Zero and not a multiple of 10, over \$200.	
Enter the following transaction to ensure proper distribution of \$20s and \$10s. (remaining 20s, remaining 10's): <ol style="list-style-type: none"> 1. \$200 – 10 \$20 bills (40, 50) 2. \$190 – 9 \$20 bills and 1 \$10 bill (31, 49) 3. \$200 – 10 \$20 bills (21, 49) 4. \$190 – 9 \$20 bills and 1 \$10 bill (12, 48) 5. \$200 – 10 \$20 bills (2, 48) 6. \$200 – 2 \$20 bills and 16 \$10 bills (0, 32) 7. \$200 – 20 \$10 bills (0, 12) 8. \$200 – Reject not enough available funds 9. \$120 – 12 \$10 bills (0,0) and end program. 	
Enter secret code to ensure correct balance of \$20s and \$10s are entered. Remaining balance and number of transactions so far.	
Does the program end gracefully with appropriate ending message and summary of transactions?	