Program with password protection

Introduction:
Areas covered in this part:
ASCII Characters, I/O, MASM Assembler, Branching, NOP operation, DEBUG commands "a" and "p",
Instructions INT 10 service 0E, INT 16 Service 00, INT 21 Function 09 and 0A.

In this laboratory assignment, you will add text input and output to a 16bit version of your final hand assembly
program. Examples of using interrupt-based services are attached and will serve as a starting point.

Part 1: Password Protection Module
Develop a program that requires that you use a two-character password and then display on the screen
the message of your choice, like, “Welcome to <insert your name> Banking program”. The program should
have the following features:
1. The password will be your personal initials (for example, the password would be “JH” for John
Hancock).
2. Initially use INT 10 service 0E to echo each character of the password to the screen immediately after
it is typed.
3. If anything but the correct password is entered, display the message, “Access Denied”.
4. After you have been checked off for Part 1 (pre-lab flow chart and demo), remove the password “echo”
to the screen by using the NOP instruction (op-code 90h). The NOP (90h) will effectively “rub out” the
password characters without changing the byte count of the program. If you just delete the code
instead of using NOPs you will have to recalculate your jumps. Or you can use “*” instead of echoing
the characters from the keyboard.
5. Modify the program to use a subroutine to check a 6 or more alpha-numeric character password

Part 2: Combining Password Protection and Calculation Modules:
Add a numeric input routine and combined programs
1. Write a routine that uses a string input INT21 function 0A to input the value of the check into the debit
location (memory address location 0202). Note: you will need to remove the “ASCII bias” from the
inputted characters. Include an input prompt message of your choice (for example: “Welcome to JN
Bank” next line “Please enter the check amount”).
2. Create a 16 bit version of your completed Hand Assembly lab program; this program will perform the
calculations (this is your calculation module). You will probably need to change your memory data
addresses and the start of your calculation module program (because part 1 password protection code
will need to start at IP 0100h). You will want to use MASM to write your code. Turn the calculation
module into a sub routine and create a method by which the program will keep count how many times
overdraft was used. Modify the program to print a title that includes your name, and displays program
results in BCD (hint; see program 5.2 to display BCD, make sure your calculations don’t go negative).

Pre-Lab Work:
1. Complete flow chart for Part 1 before the lab; show your lab instructor at the beginning of the lab.
2. Complete the flow chart for Part 2 before the lab. The flow chart will show the numeric input routine
and combined programs for Part 1 and Part 2, and show your lab instructor at the beginning of the lab.

Laboratory Work:
1. Demonstrate Part 1: Your updated pre-lab Flow Chart showing the machine instructions for each
symbol on the chart, commented code, and be prepared to discuss your program.
2. Demonstrate Part 2: Your updated pre-lab Flow Chart showing the machine instructions for each
symbol on the chart, commented code, and be prepared to discuss your program.

Lab Report Due: Week 14
Please enter current balance and then check amount

Program terminated normally

-n a:\lab2Bex.com

Please enter current balance and then check amount

Writing 00050 bytes

C:\debug a:\lab2Bex.com