**CIS 21JA - Lab 5: Branching and review of arithmetic operations, flags**

**Overview**  
Write a program that finds the time duration between a given start time and end time.  
  
**Program description**  
The program will form 3 main tasks.

1. Read the start time and end time.   
Loop 2 times to read the user input: the first loop iteration is for the start time, the second loop iteration is for the end time.   
For each loop iteration, do the following steps:

1. Prompt the user for the hour. Check that the input number is within the range of 0-23, inclusive.

If the input number is larger or smaller than the valid range, print an error message and loop back and re-prompt until you get a valid hour.

1. Prompt the user for the minutes. Check that the input number is within the range of 0-59, inclusive.

If the input number is larger or smaller than the valid range, print an error message and loop back and re-prompt until you get a valid minute.

1. After you have a valid hour and valid minute, you need to use 8-bit registers only. For this lab, we assume that your code runs on an 8-bit processor to do all the calculations.  
   Convert the hour and minute values into total minutes: total minutes = hour \* 60 + minute  
   *Note that you must use 8-bit registers only to do the calculation.*
2. During the conversion calculation, if the result is invalid (due to the 8-bit limitation), print an error message and loop back to step a to re-prompt for the hour and minutes.  
   *Note that you must use 8-bit registers only to check for valid result.*  
   Hint: Are you working with signed or unsigned data? When doing arithmetic operations, how do you know that the result is not valid?

2. Find and print the difference between the start total minutes and the end total minutes.

1. Subtract: end total minutes – start total minutes  
   *Note that the subtraction is also with 8-bit registers.*
2. With the 8-bit limitation, if the difference is invalid, print an error message and go to step 3.   
   *Note that you must use 8-bit registers only to check for valid result*  
   Hint: Are you working with signed or unsigned data? When doing arithmetic operations, how do you know that the result is not valid?
3. If the difference is valid, convert the time difference into number of hours and number of minutes (reverse of step 1c above).

*Note that you must use 8-bit registers only to do the calculation and the check for valid result*  
Hint: Are you working with signed or unsigned data? When doing arithmetic operations, how do you know that the result is not valid?

1. As above, if the result is invalid (due to the 8-bit limitation), print an error message and go to step 3.  
   Otherwise, print the number of hours and number of minutes difference (see sample output), and go to step 3.

3. Ask the user whether to continue.

1. Ask the user whether to continue and give the user a choice of: y, Y, n, or N
2. Loop back to step 1 if the user answers 'y' or 'Y'  
   End the program if the user answers 'n' or 'N'  
   Loop back and ask the user again (step 3a) if the answer is not valid

.

**Additional requirements, don't miss them:**

* Don't use any memory variables other than for text strings. Store all numeric data in registers.   
  The available registers are: EAX, EBX, ECX, EDX, EBP, ESI, EDI and their smaller sizes: AL, AH, BL, BH...
* Don’t forget to use 8-bit registers for calculations, but use 32-bit registers for IO.
* Don't use the decision directives of MASM. Implement loops and if statements with assembly instructions.
* Keep your logic flow as simple as you can. Use "fall through" logic as shown in the class notes or the book.

**Sample output**

Enter hour: 4

Enter minute: 30

The time entered is too large ; invalid during conversion to total minutes

Enter hour: 4

Enter minute: 0

Enter hour: 4

Enter minute: 12

0 hours, 12 minutes

Continue? y/n: y

Enter hour: 3

Enter minute: 20

Enter hour: 2

Enter minute: 20

Invalid time difference, check your times ; invalid during time difference calculation

Continue? y/n: y

Enter hour: 0

Enter minute: 0

Enter hour: 3

Enter minute: 0

3 hours, 0 minutes

Continue? y/n: a ; invalid answer

Continue? y/n: d

Continue? y/n: Y ; uppercase answer

Enter hour: 2

Enter minute: 0

Enter hour: 2

Enter minute: 0

0 hours, 0 minutes

Continue? y/n: n