3 September 2012 -- Computer Architectures –- part 1/2

Last & First Name, Matricola ....…...

A car rental company is offering rental rates charged according to duration and time of usage. It is required to write a 8086 assembly program to compute the duration of a rental and the costs to be charged.

Input data:
RECORD DB 4 DUP (?)
RATES DB 3 DUP (?)

Output data:
DURATION_OF_RENTAL DW (?)
COST_TO_BE_CHARGED DW (?)

Format of input data
- RECORD stores on 9 bits the starting and ending days of the year (from 1 to 366) and on 5 bits the starting and ending hours, according to the following format:
  X <starting day> <starting hour> X <ending day><ending hour>
  2   9   5   2   9   5

  with   X = unused, <starting day>, <ending day> = 1..366, <starting hour>, <ending hour> = 0..23;

  It is known in advance, that the duration of a rental is at least 1 hour. For simplicity, it is assumed that the <ending day> is greater than/equal to the <starting day> (i.e. no cross-the-year rentals are allowed).

  X<13><15>X<15><12> stands for a rental starting on the 13th day of the year at hour 15, and ending on the 15th day of the year at hour 12.

- The elements of RATES store the hourly, daily, and weekly rates in <cost units> on 8 bits, with RATES [0] → hourly rate, RATES[1] → daily rate, RATES[2] → weekly rate.

Format of output data
- DURATION_OF_RENTAL stores on 6 + 3 + 5 bits the total duration in weeks, days and hours of the rental:
  X <number of weeks of rental> <number of days of rental> <number of hours of rental>
  2   6   3   5

  X<17><2><3>stands for a rental that lasted 17 weeks, 2 days and 3 hours.

- COST_TO_BE_CHARGED stores the overall cost, expressed in <cost units> on 16 bits.

Items with corresponding points for each completed item (as uncompleted items could be not evaluated)

- Item 1. (up to 22 points) = computation of the duration of the rental (in weeks, days and hours), and the total cost assuming that the daily cost is 24 times the hourly cost and that the weekly cost is 7 times the weekly cost.

- Item 2. (up to 29 points) = computation of the duration of the rental (in weeks, days and hours), and the total cost, at the best possible (i.e. lowest) cost for the customer, under the condition that weekly cost < 7*daily cost and daily cost < 24* hourly cost. For example, if the hourly rate is 2, the daily is 11, the weekly is 50, then:

  For a rental of 0 weeks, 3 days and 5 hours, then the cost would be
  0* 50 + 3* 11 + 5* 2 = 43. In this case this is the best possible cost for the customer.

  For a rental of 0 weeks, 3 days and 6 hours, then the cost would be

Last & First Name, Matricola ....

$0* 50 + 3* 11 + 6* 2 = 45$. In this case this is NOT the best possible cost for the customer. In fact, if for the 6 hours we charge not a hourly rate, but a daily rate, then the rental becomes of 0 weeks and 4 days (no hours) and the cost is:

$0 * 50 + 4*11 = 44$, that is less than the previous value. 44 is the best possible cost for the customer.

For a rental of 0 weeks, 5 days and 3 hours, then the cost would be

$0* 50 + 5* 11 + 3* 2 = 61$. In this case this is NOT the best possible cost for the customer. In fact, if for the 5 days we charge not a daily rate, but a weekly rate, then the rental becomes of 1 week (no days, no hours) and the cost is:

$1 * 50 = 50$, that is less than the previous value. 50 is the best possible cost for the customer.

For a rental of 0 weeks, 4 days and 6 hours, then the cost would be

$0* 50 + 4* 11 + 6* 2 = 56$. In this case this is NOT the best possible cost for the customer. In fact, if for the 6 hours we charge not a hourly rate, but a daily rate, then the rental becomes of 0 weeks and 5 days (no hours) and the cost is:

$0 * 50 + 5*11 = 55$, that is less than the previous value. But still this is NOT the best possible cost for the customer. In fact, if for the 5 days we charge not a daily rate, but a weekly rate, then the rental becomes of 1 week (no days, no hours) and the cost is:

$1 * 50 = 50$, that is less than the previous value. 50 is the best possible cost for the customer.

- Item 0. (up to 4 points) = computation of tax on the rental cost and storage into the variable TAX DW (?). The tax equals 25% of the COST_TO_BE_CHARGED and has to be computed with a (the points below do not add up, but the max only is considered; therefore choose only one type out of the below):
  - trunc to cost units (up to 2 points); e.g. 12.00 →12, 12.25 → 12, 12.50 →12; 12.75 →12
  - ceil to cost units (up to 3 points); e.g. 12.00 →12, 12.25 → 13, 12.50 →13; 12.75 →13
  - round to cost units (up to 4 points) e.g. 12.00 →12, 12.25 → 12, 12.50 →13; 12.75 →13

HINT for item 2: a possible technique is:
- to compute the cost based on hours only (let's call it cost_from_hours). If this is smaller than the daily cost, then leave unchanged, otherwise reset the cost_from_hours and increase by one the number of days
- then, to compute the cost based on days and add to cost_from_hours (let's call it cost_from_days_and_hours). If this is smaller than the weekly cost, then leave unchanged, otherwise reset the cost_from_days_and_hours and increase by one the number of weeks
- then, to compute the cost based on weeks and add to cost_from_days_and_hours. This is the best possible cost for the customer.

REQUIREMENTS (SHARP)

- It is not required to provide the optimal (shortest, most efficient, fastest,…) solution, but a working and clear solution.
- It is required to write at class time a short and clear explanation of the algorithm used.
- It is required to write at class time significant comments to the instructions.
- The input-output part is not necessary in the class-developed solution, but its implementation is mandatory to be discussed at oral exam.

Please use carbon copy and retain one copy for home implementation and debug. Please provide your classroom submitted solution with several explanatory and significant comments. When coming to oral discussion, please mark on your "classroom" copy, all modifications. Please also provide an error-free and running release of the solution, as well as with its printed list of instructions. Please consider that the above are necessary but not sufficient requirements to success the exam, since the final evaluation will be based on a number of parameters.

FAILURE TO ACCOMPLISH ALL PREVIOUS NECESSARY REQUIREMENTS WILL CAUSE NO-QUESTION-ASKED AND IMMEDIATE REJECTION.