18 December 2013 -- Computer Architectures self-test –- part 1/2

Name, Matricola **.....**

It is known that each digit in base 3 ranges from 0 to 2, thus requiring 2 bits to be represented; therefore, one byte can be used to represent a value on four ternary digits. For example, $(1220)_3$ is represented as
01 10 10 00
1 2 2 0

Let us consider the arrays each one storing each one N positive 4-digits-ternary numbers
FIRST DB N DUP (?)
SECOND DB N DUP (?)

with 2<= N <= 10.

(Item 0) It is requested to produce the sorted arrays
FIRST_SORTED_HI_TO_LO DB N DUP (?)
SECOND_SORTED_LO_TO_HI DB N DUP (?)
containing, the sorted highest_to_lowest and lowest_to_highest values of FIRST and SECOND

(Item 1) It is requested to compute the addition for i=1 to 10
ADDED[i] ← FIRST_SORTED_HI_TO_LO[i] + SECOND_SORTED_LO_TO_HI[i]
with
ADDED DW N DUP (?)

(Item 2) It is the requested to compute the multiplication by three, for i=1 to 10
BY_THREE_ MULTIPLIED_FIRST[i] ← 3 * FIRST_SORTED_HI_TO_LO[i]
BY_THREE_ MULTIPLIED_SECOND[i] ← 3 * SECOND_SORTED_LO_TO_HI[i]
with
BY_THREE_ MULTIPLIED_FIRST DW N DUP (?)
BY_THREE_ MULTIPLIED_SECOND DW N DUP (?)

(Item 3) It is the requested to compute the multiplication by two, for i=1 to 10
BY_TWO_ MULTIPLIED_FIRST[i] ← 2 * FIRST_SORTED_HI_TO_LO[i]
BY_TWO_ MULTIPLIED_SECOND[i] ← 2 * SECOND_SORTED_LO_TO_HI[i]
with
BY_TWO_ MULTIPLIED_FIRST DW N DUP (?)
BY_TWO_ MULTIPLIED_SECOND DW N DUP (?)

(Item 4) It is the requested to compute the multiplication for i=1 to 10
MULTIPLIED[i] ← FIRST_SORTED_HI_TO_LO[i] * SECOND_SORTED_LO_TO_HI[i]
with
MULTIPLIED DW N DUP (?)

ALL OPERATIONS SHOULD BE CARRIED OUT IN BASE 3, i.e. without passing through base 2 or other bases.

HINT: especially for multiplication, before coding the instructions write down the Pythagorean table and write a procedure/portion of code, to implement it; do the same also for addition (see examples below).

The points related to writing correct 8086 assembly code for each completed item (as uncompleted items will not be evaluated)

- Item 0 (**MANDATORY**): 10 points
- Item 1: 10 points
- Item 2: 3 points
- Item 3: 6 points
- Item 4: 20 points

18 December 2013 -- Computer Architectures self-test –- part 1/2

Name, Matricola **.....**

Please consider that a maximum of 33 points can be accounted here; larger values will be "cut" to 33.

Example of addition: $(1201)_3+(1102)_3$ = … let's start from right to left:
1+2 makes **0** with a carry of 1
0+0+carry of 1 makes **1** with a carry of 0
2+1+carry of 0 makes **0** with a carry of 1
1+1+carry of 1 makes **0** with a carry of **1**
therefore, $(1201)_3+(1102)_3 = (10010)_3$

Example of multiplication by two: $2*(1201)_3$ = … let's start from right to left:
2*1 makes **2** with a carry of 0
2*0+carry of 0 makes **0** with a carry of 0
2*2+carry of 0 makes **1** with a carry of 1
2*1+carry of 1 makes **0** with a carry of **1**
therefore, $2*(1201)_3 = (10102)_3$

REQUIREMENTS (SHARP)

- **Prior solving other items, it is MANDATORY to solve Item 0**
- 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.
- Minimum score to "pass" this part is 15 (to be averaged with second part and to yield a value at least 18)

REQUIREMENTS ON THE I/O PART TO BE DONE AT HOME

- The database has to be defined and initialized inside the code
- The program should present a menu with the choices corresponding only to the items developed during the written exam
- All inputs and outputs should be in readable ASCII form (no binary is permitted).

Please use carbon copy ONLY (NO PICTURES ARE ALLOWED) and retain one copy for home implementation and debug. At the end of the exam please give to professors all the sheets of your solution. Missing or late sheet will not be evaluated. Please provide your classroom submitted solution with several explanatory and significant comments. Please remember that only what has been developed at class time can and will be evaluated at oral time and that it is necessary to write the instructions of the program and not just the description of the algorithm.

When coming to oral discussion, please clearly mark in red 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 THE ABOVE NECESSARY REQUIREMENTS WILL CAUSE NO-QUESTION-ASKED AND IMMEDIATE REJECTION.