23 February 2015 -- Computer Architectures –- part 1/2

Matr, Last Name, First Name **……**

Nowadays all cars are equipped with an electronic dashboard providing the driver with useful information tools to track the usage of the car. Among the most common functions offered there are the visualizations of the following parameters since their last reset: Number of Km driven, duration of the drive, number of Km with one liter of fuel, number of deciliters per 100 Km, average speed, number of liters still in the tank, number of drivable Km with the available fuel in the tank.
It is assumed that: the fully refilled tank contains 32 liters of fuel; the counters are reset each time the tank is completely refueled; the tank should be refueled at most every 4 hours of travel; at the first run of the program the tank is full and all other parameters are set to zero (for the number of Km drivable with the available fuel in the tank it is by convention assumed that it is 800).
It is also assumed that ranges of the different parameters are as follows (no need to check by the program):
overall number of Km driven ∈ [0, 800],
overall duration of the drive (in minutes) ∈ [0, 240],
overall number of Km with one liter of fuel ∈ [0, 30],
overall number of deciliters per 100 Km ∈ [0, 250],
overall average speed (in Km/hour) ∈ [0, 200],
number of deciliters still in the tank ∈ [0, 320],
overall number of further drivable Km with the available fuel in the tank ∈ [0, 800].

It is requested to write an 8086 assembly program to emulate a car dashboard. It is assumed that from time to time the program receives an update on the travel parameters, consisting in incremental values since the previous update, i.e.: number of deciliters burned since the last update (on 8 bits), number of Km driven since the last update (on 8 bits), number of minutes since the last update (on 6 bits), in the variable UPDATE DB 3 DUP (?) according to the format:

ddddddddd	KKKKKKKK	00mmmmmm
deciliters	Km driven	minutes

For simplicity, the user will provide the update through the keyboard at runtime. The final program will have to loop and a menu has to be provided to the user with the following options:
1) enter an update (deciliters burned; Km driven, minutes driven)
2) display the current available parameters
3) completely refuel the tank and reset all parameters
4) exit the program

- **(mandatory)** Item 1: the program has to compute and display (by request): overall number of Km driven, overall duration of the drive in minutes, overall used deciliters of fuel, number of **liters** still in the tank (truncated from the fractional part);
- Item 2: to compute the overall average speed in **Km/hour** as an integer value obtained by truncation with the **maximum possible precision**
- Item 3: to compute the overall duration of the drive **in hours and minutes**
- Item 4: to compute the number of **Km/ liter** as an integer value obtained by truncation with the **maximum possible precision**
- Item 5: to compute the number of **deciliters per 100 Km** as an integer value obtained by truncation with the **maximum possible precision**
- Item 6: to compute the number of **further drivable Km with the available fuel in the tank** (based on previous fuel usage and Km driven) as an integer value obtained by truncation with the **maximum possible precision**

The points related to writing correct and completed items (as uncompleted items will not be evaluated) are:
- Item 1: 22 points
- Item 2-5: each one 2 points
- Item 6: 3 points
- For three Items 2-6 correctly solved -> 1 point more;

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

23 February 2015 -- Computer Architectures –- part 1/2

Matr, Last Name, First Name **……**

HINTS

- To achieve at best the maximum precision, the domains of the variables should be carefully considered. For example, for computing the speed S in Item 2, it is necessary to have available the number of Km driven (K) and the overall duration of the trip (in minutes, D). As it is requested to compute Km/hour with the maximum precision it is necessary to compute S=K/(D/60). However, if H=D/60 is computed first via an integer division, then the precision achieved is not the best. If we compute S=(K*60)/D we do not face this problem. But how many bits? K, according to its range defined above, needs 16 bits; D requires 16 bits; S requires 8 bits. Therefore K*60 will be a 16 bits x 16 bits multiplication producing a result on 32 bits which is then divided D, i.e. a 32 bits / 16 bits division. The result of the division will be on 16 bits (integer part only), where only the least significant byte represents the result S.
- For each item, before going to the implementation, **Students are (mandatory) requested** to write down an analysis similar as this above for Item 2, and then proceed with the implementation.
- Do *NOT* derive **deciliters per 100 Km** from **Km/ liter** or viceversa; same also for **further drivable Km with the available fuel in the tank has** not to be drived from **Km/ liter** and **liters** still in the tank. Due to truncation errors these results will not be correct, i.e will not have the maximum possible precision.

Example: INPUT: deciliters=54 Km driven=80 minutes=45
DISPLAY: overall number of Km driven= 80; overall duration of the drive in minutes=45;
overall used deciliters of fuel= 54; number of **liters** still in the tank (truncated from the fractional part)=26;
average speed in **Km/hour** = 106; overall duration of the drive **in hours and minutes** = 0h 45m; **Km/ liter** = 14;
deciliters per 100 Km = 67; **further drivable Km with the available fuel in the tank** = 394 (and not 14+26=364)
+++
INPUT: deciliters=15 Km driven=32 minutes=20
DISPLAY: overall number of Km driven= 112; overall duration of the drive in minutes=65;
overall used deciliters of fuel= 69; number of **liters** still in the tank (truncated from the fractional part)=25;
average speed in **Km/hour** = 103; overall duration of the drive **in hours and minutes** = 1h 05m; **Km/ liter** = 16;
deciliters per 100 Km = 61; **further drivable Km with the available fuel in the tank** = 407 (and not 16*25=400)
+++
INPUT: deciliters=44 Km driven=61 minutes=77
DISPLAY: overall number of Km driven= 173; overall duration of the drive in minutes=142;
overall used deciliters of fuel= 113; number of **liters** still in the tank (truncated from the fractional part)=20;
average speed in **Km/hour** = 73; overall duration of the drive **in hours and minutes** = 2h 22m; **Km/ liter** = 15;
deciliters per 100 Km = 65; **further drivable Km with the available fuel in the tank** = 316 (and not 15*20=300)

REQUIREMENTS (SHARP)

- It is not required to provide the optimal (shortest, most efficient, fastest) solution, but a working and clear one.
- 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
- 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.