5 September 2016 -- Computer Architectures –- part 1/2

Matr, Last Name, First Name **……**

Triangle ABC has the basis AB lying on the line y=(3/4)x and point C at coordinates x_C, y_C and is completely in the positive quadrant. It is requested to compute the area of the triangle ABC using the well-known formula for computing the distance of a point from a line, i.e.:

the distance of point P (x_0, y_0) from the line y=mx+q is given by $d = |\frac{y_0 - mx_0 - q}{\sqrt{1+m^2}}|$

It is requested to write an 8086 assembly program to compute the area of the triangle, given the coordinates of A, B and C, or the length AB and the coordinates of C.

Each coordinate is stored on 8 integer bits in binary representation. The area should be computed on 16 integer bits with the maximum as possible precision neglecting the fractional part (if any); no rounding is requested.

Tasks to be implemented and corresponding point (only fully completed items will be considered to award points); only one among Items A, B, C should be solved.

- Item A: assume that the basis length is 30, i.e. AB = 30, and x_C, y_C are provided at run time; POINTS → 21
- Item B: assume that AB = 12 and x_C, y_C are provided at run time; POINTS → 24
- Item C: assume that x_A, x_B, x_C, and y_C are provided at run time; POINTS → 31
- **Bonus Item**: given a pair of coordinates x_T, y_T with x_T multiple of 4, determine if point T belongs to the line y=(3/4)x POINTS → +3.

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

Depending on the Item solved, the program should receive in input the values of the parameters and should display the final value of the area, computed with precision, but neglecting any rounding issue as well as any fractional bit.

A few examples:
Item A: AB=30; x_C= 1 y_C= 1 → area = 3
Item A: AB=30; x_C= 2 y_C= 1 → area = 6
Item A: AB=30; x_C= 2 y_C= 2 → area = 6

Item B: AB=12; x_C= 2 y_C= 2 → area = 2 (with some truncation, which should not be computed)
Item B: AB=12; x_C= 1 y_C= 2 → area = 6 (without any truncation)
Item B: AB=12; x_C= 3 y_C= 1 → area = 6 (without any truncation)
Item B: AB=12; x_C= 4 y_C= 3 → area = 0 (without any truncation)

Item C: x_A= 1 x_B= 3 x_C= 1 y_C= 3 → area = 2 (with some truncation, which should not be computed)
Item C: x_A= 0 x_B= 6 x_C= 2 y_C= 3 → area = 4 (with some truncation, which should not be computed)
Item C: x_A= 4 x_B= 5 x_C= 0 y_C= 6 → area = 3 (without any truncation)
Item C: x_A= 9 x_B= 5 x_C= 1 y_C= 6 → area = 10 (with some truncation, which should not be computed)

HINTS (observe that)

- **This is a hard-to-think & easy-to-code algorithm. Please spend some time to compute, by hand, the area of the triangle, based on ½ AB*d where d is given by the formula above, in terms of AB length and x_C, y_C coordinates.**
- Please observe that as the value of m is already known (i.e. m=3/4), in the distance formula it is not necessary for the program to compute a square root value, as it can be computed offline, i.e. by the analytical analysis. For the assembly program to be written, $\sqrt{1+m^2}$ is a predefined constant whose value has to be computed by the programmer by hand while performing the preliminary analysis of the problem.
- Students affording problem in Item C, are gently invited to find first the formula for the length of AB depending on x_A, x_B. Due to the particular property of AB to lying on the line y=(3/4)x, Students will easily observe that there is no need for the program to compute any square root value.
- The value of the area is a positive number!
- It is advised to design the program as a collection of modules, each one implementing the different Items.

5 September 2016 -- Computer Architectures –- part 1/2

Matr, Last Name, First Name **…….....…….....**

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.
- Input-output is not necessary in class-developed solution, but its implementation is mandatory for the 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 databases (if any, i.e. not necessary in case) have 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.