8 September 2015 -- Computer Architectures –- part 1/2

Matr, Last Name, First Name **……**

(Credits to wikipedia.org) Bowling is a game in which a player rolls a bowling ball towards a target. The bowler is allowed 10 frames in which to knock down 10 pins, with frames (1) through (9) being composed of one roll (in case of strike) or two rolls (in all the other cases). The 10th frame is composed by two (if spare or strike are not present) or three rolls (in case of strike in the first roll or spare in the second roll), i.e. the bonus roll(s) following a strike or spare in the 10th are fill ball(s) used only to calculate the score of the mark rolled in the 10th. Bowling has a unique scoring system, which keeps track not only of the current pinfall in a frame, but also strikes and spares, which allow for the value of subsequent pinfall. Effectively, there are three kinds of marks given in a score:

- a strike (all ten down in the first ball); a strike earns ten points plus the points of the next two balls thrown (usually in the next frame, but possibly in the two next frames, in case of two consecutive strikes); e.g.: a strike followed with a 7 and 2 in next rolls→ their value for the strike frame would be 10+7+2, i.e., 19.
- a spare (all ten down by the second ball); a spare earns ten points plus the points for the next ball thrown; e.g.: a spare followed with a 7 and 2 in next rolls → their value for the spare frame would be 10+7, i.e., 17.
- an open (some missed pins still standing after the second ball); they count the value of the pinfall in the frame.

The maximum score in ten-pin bowling is 300 and consists of getting one strike each in frames 1–9, and all three possible strikes in the 10th frame (strike in the first roll of the 10th frame plus two strikes in the two bonus rolls), each accounting 30 points for each frame.

It is requested to write an 8086-assembly program to compute the scores of an **8-frames-only** bowling game called **POLIBOW**, where **8 pins** only have to be hit. For **POLIBOW** the maximum score is 24*8=192. The program:

1. displays the number of frame, from one to eight and for each frame:
    a. displays the number of ball roll;
    b. receives from the user the number of pins that have been hit; this number is from 0 to 8;
    c. according to the number of roll and hit pins determines the next action, i.e., second roll or frame end;
2. at the end of a frame, displays the number of pins hit during the first and second roll (if any, with "-" if there has not been a second roll), the letter "e" for a "spare" if achieved, the letter "s" for a "strike" if achieved, or "n" for nothing, and the full list of points achieved, frame by frame up to the current frame as well as the total number of points achieved so far. Clearly, the 8th frame should be considered differently, as it could have up to 3 rolls (at worst split into two plus one extra). Clearly, if a strike has been obtained at a frame, it is not possible to exactly display the points for that frame, as also the next two rolls will contribute. In this case the displayed points will be the real points followed by two "?" informing that two more rolls will be necessary. For a spare, the real points will be the frame actual points followed by one "?" only
3. at the end of the last frame (which will be made up to 2+1 rolls) the full picture of the game, frame by frame, and the total number of points should be provided.

Please remind that POLIBOW uses only 8 pins, i.e. strike or spare occur when 8 pins are hit.

Frame	First roll	Second roll	Displayed: frame number, 1st,2nd roll, spare/strike/nothing, frame p.ts, total p.ts
1	0	8	frame #1: 0, 8, e, frame points = 8?, total points = 008?
2	5	2	frame #1: 0, 8, e, frame points = 13, total points = 013 frame #2: 5, 2, n, frame points = 7, total points = 020
3	8	-	frame #1: 0, 8, e, frame points = 13, total points = 013 frame #2: 5, 2, n, frame points = 7, total points = 020 frame #3: 8, -, s, frame points = 8??, total points = 028??
4	3	5	(frames #1 to #2 are displayed as above) frame #3: 8, -, s, frame points = 16, total points = 036 frame #4: 3, 5, e, frame points = 8?, total points = 044?
5	8	-	(frames #1 to #3 are displayed as above) frame #4: 3, 5, e, frame points = 16, total points = 052 frame #5: 8, -, s, frame points = 8??, total points = 060??
6	8	-	(frames #1 to #4 are displayed as above) frame #5: 8, -, s, frame points = 16?, total points = 068? frame #6: 8, -, s, frame points = 8??, total points = 076???
7	1	2	(frames #1 to #4 are displayed as above) frame #5: 8, -, s, frame points = 17, total points = 069 frame #6: 8, -, s, frame points = 11, total points = 080 frame #7: 1, 2, n, frame points = 3, total points = 083

8 September 2015 -- Computer Architectures –- part 1/2

Matr, Last Name, First Name **……**

8	5	3	(frames #1 to #7 are displayed as above) frame #8: 5, 3, e, frame points = 8?, total points = 091?
Extra	8		(frames #1 to #7 are displayed as above) frame #8: 5, 3, e, frame points = 16, total points = 099 ** extra = 8

TASKS: Students are requested to COMPLETELY solve EITHER Item 1 or 2 or 3; Unfortunately, "incremental/upgraded" (e.g. Item 1 + the instructions to upgrade it to item 2) or "partial" solutions will NOT be accepted

- Item 1: the program operates according to specifications above, assuming that no strike can be achieved overall and no spare in frame 8 (spare can occur in frames 1-7);
- Item 2: the program operates according to specifications above, and strikes can be achieved in all frames except 8, as well as spares can be achieved in all frames including 8;
- Item 3: operates according to specifications above, and strikes can be achieved in all frames including 8, as well as spares can be achieved in all frames including 8;
- Bonus: the following statistics are computed:
    - Maximum points in a frame (including number of frame)
    - Minimum point in a frame (including number of frame)
    - Average points during all frame (it is forbidden to use a DIV instruction)

The points related to writing correct and completed items (as uncompleted items will not be evaluated) are:

- Item 1: 22 points
- Item 2: 26.5 points
- Item 3: 31 points
- Bonus: 1 points per statistics

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

HINTS (observe that)

- Total points is a positive number not larger than 192, i.e. fitting 8 bits
- In the "worst" case, at most three frames can present some "?" i.e. uncompleted scores based on future rolls
- The key of the problem is to design a proper database and to be very clear in the algorithm flow chart; my proposal is to have for each frame a suitable number of cells, including some "extra" cells for holding the hit values necessary to complete the score for that frame

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 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 <u>all</u> 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 <u>in red</u> on your "classroom" copy, <u>all modifications</u>. 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.