4 July 2011 -- Computer Architectures -- part 1/2

Name, Matricola ....

The airline company SimpleJet allows each passenger to check in 2 luggage pieces (denoted as 0 and 1) provided that:

- the weight of the items in each luggage is not exceeding 24 Kg
- the weight of the checked items (into luggage 0 + 1) is not exceeding 40 Kg

As a passenger of SimpleJet you have to decide how to pack your items in the two allowed pieces. You have 12 items (tagged from 0 to 11) with corresponding weight value according to the list WEIGHTS DB 12 DUP(?), where each weight is represented in fixed point according to the following notation: pure binary on 6 integer + 2 fractional bits, i.e. xxxxxx.xx .

It is known in advance that the overall weight of all the items does not exceed 63 Kg.

It is required to write a 8086 assembly program to determine how the different items have to be put into one piece or the other, in order to meet the check in requirements of SimpleJet. Clearly, the problem is symmetrical, as the roles of luggage 0 and luggage 1 can be swapped.

In particular, it is required that:

- Item 0: a check is performed about the maximum weight of items that have to be carried, vs. SimpleJet rules;
- Item 1: one possible solution of packaging the items is found and printed, according to the rules (and assuming that at least one solution exists)
- Item 2: all possible solutions are found and printed, as well also the management of the case where no possible solution exists
- Item 3: one of the "best solutions" is found and printed, where for "best solution" we mean a solution where the weight difference between the two pieces is the smallest as possible
- Item 4: all "best solutions" are found and printed, where all of them have exactly the same difference of weights between the two pieces.

Bonus item: the symmetry of the problem and solution is managed and are therefore not presented as different the solutions where the roles of luggage 0 and 1 are simply swapped.

It is required that a solution is internally represented (and printed as well) in a word according to the following format:

0000 XXXX XXXX XXXX

where each X is a 0 if the corresponding "column numbered piece" has been stored in piece 0 (first luggage), and a 1 if it has been stored in piece 1 (second luggage). The underscored to four zeroes are just to fill the word length and do not refer to any item

For example:

```
                this is item number 0
                              !
0000 0011 0001 1001
       ^
this is item number 11
```

is coding that into piece 0 have been stored the items with a "0" in the previous word, i.e.: item 1, item 2, item 5, item 6, item 7, item 8, item 10, item 11. In addition, into piece 1 there are the following items: item 0, item 3, item 4, item 8, item 9.

4 July 2011 -- Computer Architectures –- part 1/2

Name, Matricola ....

Points for each completed item (as uncompleted items could be not evaluated)
Item 0. = up to 1 point;
Item 1. = up to 20 points;
Item 2. = up to 2 points;
Item 3. = up to 5 points;
Item 4. = up to 3 points;
Bonus = up to 1 point.

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

IMPORTANT NOTE:
Theoretically, the problem can be solved either with recursion, but there exist also very simple and straightforward solutions based on iterations only. The notation that has been recommended in the text is particularly useful in the case of designing an iterative solution.

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.