Microprocessor Based Systems. Exam of February 19, 2010.

A passenger plane has the economy class on 50 rows of 8 seats, and the business class on 10 rows of 4 seats each one. Rows are labelled with numbers from 1 to 10 for business and from 11 to 60 for economy. Columns are labelled with letters and the layout of the economy class is
AB CDEF GH (A and H are window seats, and B,C,F,G are aisle seats)

while the layout of business class is
AB GH (A and H are window seats, and B and G are aisle seats).

In the database of the air company, associated to each seat there is the information about its occupancy:
00 = to be assigned
11 = unavailable
01 = occupied by a man
10 = occupied by a woman

The information related to one row in economy is packed into a single word where the two most significant bits are related to seat A, the next to seat B and so on up to the least significant which are related to seat H. For example, the row coding
01 00 00 00 11 10 10 01 stands for the following column occupation (for that row):

A,H = occupied by a man
B,C,D = to be assigned
E = unavailable
F,G = occupied by a woman

In business class the seats occupation is coded with the same couple of bits but is packed into a byte.
The database of the plane is stored with the rules above by the two arrays:
ECONOMY DW 50 DUP (?) and BUSINESS DB 10 DUP (?)

It is asked to write an assembly 8086 program for doing a simple database management, according to the operations which are usually performed by a check-in desk. The operations to be handled are listed below and have to be done each one after the other, i.e. without looking for a global optimization but on a first-come-first-served basis:

Code 0= quit the seat assignment procedure

For economy class:
Code 10= window seat preference, possibly close to a woman
Code 11= window seat preference, possibly close to a man
Code 12= window seat preference
Code 20= aisle seat preference, possibly close to a woman
Code 21= aisle seat preference, possibly close to a man
Code 22= aisle seat preference
Code 33= generic seat assignment, i.e. take one seat with no preferences, among these “to be assigned”

For business class:
Code 110= window seat preference, possibly close to a woman
Code 111= window seat preference, possibly close to a man
Code 112= window seat preference
Code 120= aisle seat preference, possibly close to a woman
Code 121= aisle seat preference, possibly close to a man
Code 122= aisle seat preference
Code 133= generic seat assignment, i.e. take one seat with no preferences, among these “to be assigned”

The order of preference satisfaction is first try to accommodate the position (window/aisle) and then the proximity (woman/man). The parameter “global seat assignment satisfaction” is valued in points, according to the following rules: a satisfaction in position (window/aisle) is valued 3 points, while a satisfaction in proximity (man/woman) is valued 2 point. A generic seat assignment without preferences is valued 1 point. A seat assignment with no satisfaction on all requests is valued 0 points.

The program should as a first step get the initial value of the database ECONOMY and BUSINESS, by using some input method. While getting the database, the program should count the number of available seats. Then the program should enter the seat assignment procedure, assuming by definition the number of requests of seats does not exceeds the number of seats still available, and that no wrong code is entered by the check-in desk operator.

In correspondence of each request, the program should try to accommodate it at its best with respect to the current situation of seats assigned so far, and should:

- immediately print on the screen the code of the seat which was assigned, i.e. column letter and row number (e.g. C25)
- update the value of the “global seat assignment satisfaction” (without backtracking on previous seat assignments)
- keep track both of the seat assignment (in order not to assign the same seat to another passenger), and of the free seats still to be assigned.

At the end of operations, either because all free seats have been assigned or because a code zero has been entered (i.e. quit the seat assignment procedure), the program should:

- provide the correct value of the database ECONOMY and BUSINESS, according to seats assignments
- print in output the value of the parameter “global seats assignment satisfaction”
- print in output the number of seats which have been assigned by the procedure, as well as the number of seats still not assigned (i.e. free).

The program to be developed in class-phase of the exam can assume that initial ECONOMY and BUSINESS values are known in advance. Then, it has to focus on the instructions necessary to implement the features listed above. While developing their program in class, students can make reference to a generic PRIN_T(X) procedure for outputting on the screen the value of the variable(s) X. The final version to be discussed and verified should implement each printing function. For example, PRIN_T (column, row) can be used in the exam solution, provided that row and column variables are computed by instructions above the PRIN_T. The proper instructions for good printing have to be implemented in the version of the program which is prepared for oral discussion.

Please use carbon copy and retain one copy for home implementation and debug. Please provide in your classroom submitted version several explanatory and significant comments.