14 February 2013 -- Computer Architectures –- part 1/2

Last and First Name, Matr. ....

To control the weight it is necessary to balance the phases of getting and burning calories. Let us assume to have two tables, Nutrition and Tasks, containing information about calories received by eating some food and consumed by doing some activities, respectively. Let us to have a record of the actions of a person during one day (food eating and tasks performed). It is necessary to write a program in the assembly language to compute the balance of the calories gotten and burned and to recommend proper corrective actions, i.e. eating or doing some activities.

In particular:

- Array NUTRITION DB NUTR DUP (?) contains the detailed list of calories provided by each single food or beverage provided by 25 grams, according to the following format: xxxxxxxx = binary integer representation (B.I.R.) of the number of calories.
- Array TASKS DB ACT DUP (?) contains the detailed list of calories burned by each single activity lasting 5 minutes, according to the following format: xxxxxxxx = B.I.R. of the number of calories burned.
- Array BIAS_GENDER DB 2 DUP (?) contains the number of calories that are necessary for static survival of a man (index 0) or a woman (index 1), according to the following format: xxxxxxxx = B.I.R. of the number of calories used divided by 2^3; for example 00000111 corresponds to 7*8 = 56 calories (used).
- Array PERSON DW REC DUP (?) contains the gender and the records of the activities during one day. The format is as follows:
  - 00xxxxxx xxxxxxxx = person is gender Male
  - 01xxxxxx xxxxxxxx = person is gender Female
  - 10 xddddd xiiiiiii = nutrition item; ddddd represents in binary the weight in multiple of 25 grams; iiiiiii represents in binary the code of the nutrition element inside array NUTRITION
  - 11 xddddd xiiiiiii = task item; ddddd represents in binary the duration in multiple of 5 minutes ; iiiiiii represents in binary the code of the task type inside array TASKS
- It is known in advance that in the array PERSON, there is always one and only one item related to the gender
- It is know in advance that 5 <= NUTR <= 100; 5 <= TASKS <= 100; and 4 <= REC <= 120.
- It is known in advance that no more than 6000 calories can be gotten or consumed by a person (in a day), including the “gender calories”.

Given the above arrays, the program should:

- Item 1: compute the total amount of calories that have been gotten by a person, as stored in some items of the array PERSON
- Item 2: compute the total amount of calories that have been consumed by a person according to the tasks that he/she has taken, plus the “gender” calories (all this information is stored in some items of PERSON)
- Item 3: compute the excess/lack of calories, as a two’s complement integer number of width sufficient to represent the “worst” cases.
- Item 4: in case of excess of calories, list a set of activities as well their duration in multiples of 5 minutes, in order to reduce the excess. The list should be in the same format as items in PERSON and should consist of at least 2 elements aimed to the reduction of the number of excess calories by at least 50%, still leading to a final balance greater than/equal to 0 (not the optimal, but just a solution!). The program should also identify the cases when no solution exists with the current items in TASKS.
- Item 5: in case of lack of calories, list a set of nutrition elements as well their amount in multiples of 25 grams, in order to reduce the lack. The list should be in the same format as items in PERSON and should consist of at least 2 elements aimed to the reduction of the number of lacking calories by at least 50%, and still leading to a final balance smaller than/equal to 0 (not the optimal, but just a solution!). The program should also identify the cases when no solution exists with the current items in NUTRITION.

The scores for each item are:

- Item 1.: up to **9 points**
- Item 2.: up to **10 points**
- Item 3.: up to **2 points**
- Item 4. OR Item 5.: up to **8 points**
- Item 4. PLUS Item 5.: up to **12 points**

14 February 2013 -- Computer Architectures –- part 1/2

Last and First Name, Matr. ....

EXAMPLE.

item	NUTRITION (per 25 grams)	item	TASKS (per 5 minutes)	BIAS_GENDER	calories burned (static surv.)
0	101	0	31	0 (male)	175 → 175*8=1400 calories
1	20	1	27	1 (female)	125 → 125*8=1000 calories
2	120	2	100		
3	33	3	72		
4	85	4	61		

item	PERSON	
0	00xxxxxx xxxxxx	Gender Male; **consumed 1400 cal**
1	10x00110 x00011	10 = *nutrition* item; 00110 = 6*25 grams; 00011 item number = 3; *gotten* **6*33 = 198 cal**
2	11x00100 x00001	11 = **task** item; 00100 = 4*5 minutes; 00001 item number = 1; **consumed 4*27 = 108 cal**
3	10x01011 x00010	10 = *nutrition* item; 01011 = 11*25 grams; 00010 item number = 2; *gotten* **11*120 = 1320 cal**

The results are:

- Total amount of calories gotten is: 198+1320 = 1518
- Total amount of calories consumed is 1400+108 = 1508
- Excess of calories = 1518-1508 = 10
- As 10 is smaller than 27 which is the smallest number of calories burning inside the table TASKS, it is not possible to find activities in tasks that can help to reduce this excess.

REQUIREMENTS (SHARP)

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

- Arrays NUTRITION, TASKS and BIAS_GENDER have to be defined and initialized inside the code, and both NUTRITION and TASKS should contain at least 22 items each one.
- Array PERSON must be filled at run time through some I/O procedure, asking to input the necessary data (gender, nutrition items, task items) in ascii-user-readable-form, e.g. N, 3, 12 stands for: Nutrition item number 12 with weight = 3*25 grams. Then, the program has to convert each line data to the internal format of array PERSON which will then be used by the core of the program that has been developed at exam time.
- For Items 1-5 (developed at exam time) the program should display the results in ascii form, including proper messages in case no solution exists for Items 4 and 5.

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.