BİL 102 – Computer Programming
HW 06

Last Submission Date: April 7, 2015 – 11:00am

For Questions and Grades about the HomeWork#6 see T.A. Meltem Çetiner
(e-mail: mcetiner@bilmuh.gyte.edu.tr Office No: 118 - Medical Tech Lab)

1.**(50 Pts)** Write a complete C program for job assignment problem. The aim of this program is to assign jobs to the employees as fair as possible and show the most hardworking employee of the day and the week. There are 4 employees; Ali, Ayşe, Fatma, Mehmet. They work every day of the week. You will assign one job each day to each employee for seven days. Therefore, 28 jobs to be assigned to employees. Each job has an energy value indicating its hardness. The energies of the jobs are given in a text file called "**Energies.txt**". In this file, the jobs of the same day are in the same line. So it includes seven lines and four columns as in the figure 1. When assigning jobs of a day, consider all previous assignments to all employees such that the job requiring the biggest energy is assigned to the employee with lowest total energy for the previously assigned jobs. For example, in the assignment in figure1, for Friday, total energy for previously assigned jobs are 250 (80+30+100+40), 230, 220, 210 for Ali, Ayşe, Fatma and Mehmet respectively. Energies of the jobs on Friday are 50, 40, 80 and 5. So, for justice, hardest job (requiring energy of 80) is assigned to the one with lowest "total energy for previously assigned jobs", Mehmet, and also second hardest to the second lowest and so forth.

The implementation details of your program should be as follow;

•Define an enumerated type for employees and another one for days of the week.

•You have to read the energy of the jobs from the file "**energies.txt**", create the matrix **job_energies**[**NUM_EMPLOYEES**][**NUM_DAYS**] and fill it in.

void read_matrix(const char* file_name, int m[NUM_EMPLOYEES][NUM_DAYS])

•Create the job schedule matrix **schedule[NUM_EMPLOYEES][NUM_DAYS]** that shows the energy values for each day and for each employee. For this, you use the following function:

void create_work_plan(int job_schedule[NUM_EMPLOYEES][NUM_DAYS], int m[NUM_EMPLOYEES][NUM_DAYS]) ;

•Fill the job schedule matrix s. You assign the job that is need the biggest energy to the employer who spent less energy than the others.

•Finds the most hardworking person of the given day and returns the employer.

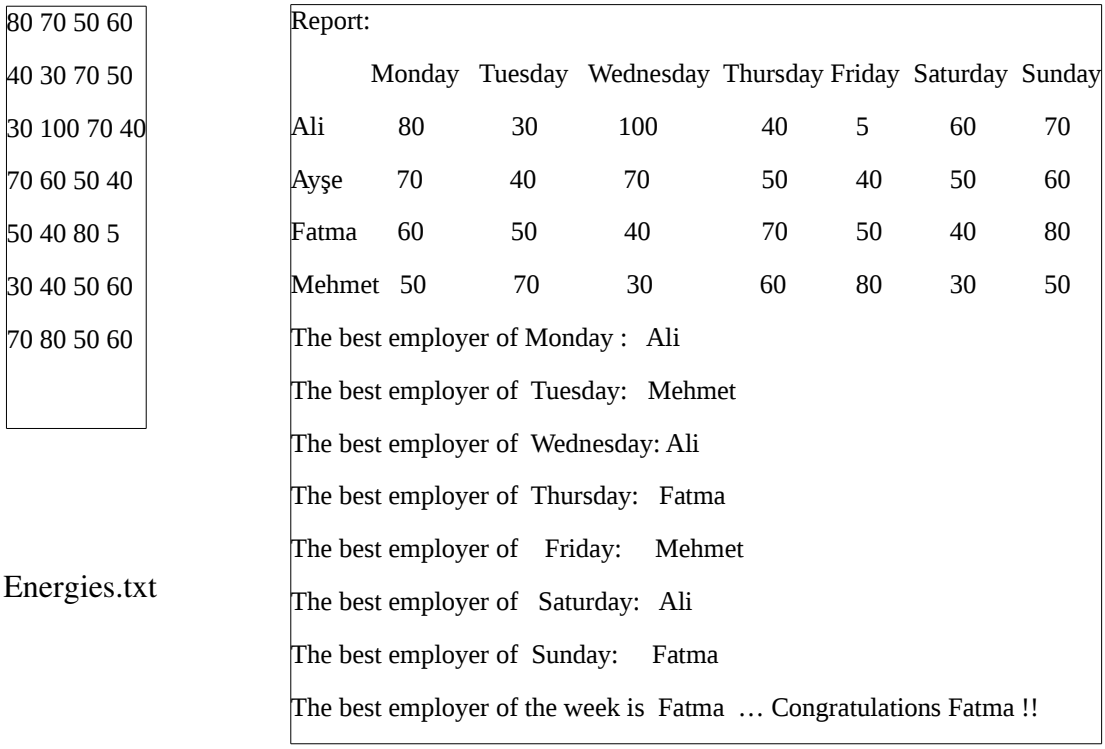
employee find_the_employer_of_the_day(int work_schedule[NUM_EMPLOYEES] [NUM_DAY], day day_name)

•Find the best employee of the week

employee find_the_employer_of_the_week(int work_schedule[NUM_EMPLOYEES] [NUM_DAY])

- prints the report to the file named "**report.txt**" as sample.

void report(const char* file_name, int job_scheduling[NUM_EMPLOYEES][NUM_DAYS])



80 70 50 60
40 30 70 50
30 100 70 40
70 60 50 40
50 40 80 5
30 40 50 60
70 80 50 60

Energies.txt

Report:

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Ali	80	30	100	40	5	60	70
Ayşe	70	40	70	50	40	50	60
Fatma	60	50	40	70	50	40	80
Mehmet	50	70	30	60	80	30	50

The best employer of Monday : Ali
The best employer of Tuesday: Mehmet
The best employer of Wednesday: Ali
The best employer of Thursday: Fatma
The best employer of Friday: Mehmet
The best employer of Saturday: Ali
The best employer of Sunday: Fatma
The best employer of the week is Fatma … Congratulations Fatma !!

Report.txt

Figure 1. Input ,and Output files of question 1

2. (50 Pts) Write a C program for "**checking the Major and Minor Vowel Harmony and making a noun Plural**". You can remember the rules of the vowel harmony using the links below. http://www.turkishclass.com/turkish_lesson_59 , http://www.turkishclass.com/turkish_lesson_60

First, you will read the vowels from the file called "**Vowels.txt**". Assume that the file consists of 4 lines of vowels that include hard, soft, flat and round vowels, respectively. So, in Turkish we expect the file to be as follows:

aıou

eiöü

aeıi

oöuü

- You use two enumerated types **major_type** for constants HARD, SOFT, CONSONANT, **minor_type** for constants FLAT, ROUND, CONSONANT and **bool** for constants TRUE and FALSE.
- The file names should be defined as constant macros.

Your program should contain at least 6 functions :

– **bool is_major_vh_word(const char* word, const char* v_hard, const char* v_soft)** checks whether the word satisfies the major vowel harmony or not and returns TRUE or FALSE. Use the following function in the implementation of this function.

– **major_type major(const char ch1, const char* v_hard, const char* v_soft)** takes one character and two lists of hard and soft vowels and checks whether the character is a soft vowel or a hard vowel. It should return HARD, SOFT or CONSONANT.

– **bool is_minor_vh_word(const char* word, const char* v_flat, const char* v_round)** checks whether the word satisfies the minor vowel harmony or not and returns TRUE or FALSE. Use the following function in the implementation of this function.

– **minor_type minor(const char ch1, const char* v_flat, const char* v_round)** takes one character and two lists of flat and round vowels and checks whether the character is a soft vowel or a hard vowel. It should return FLAT, ROUND or CONSONANT.

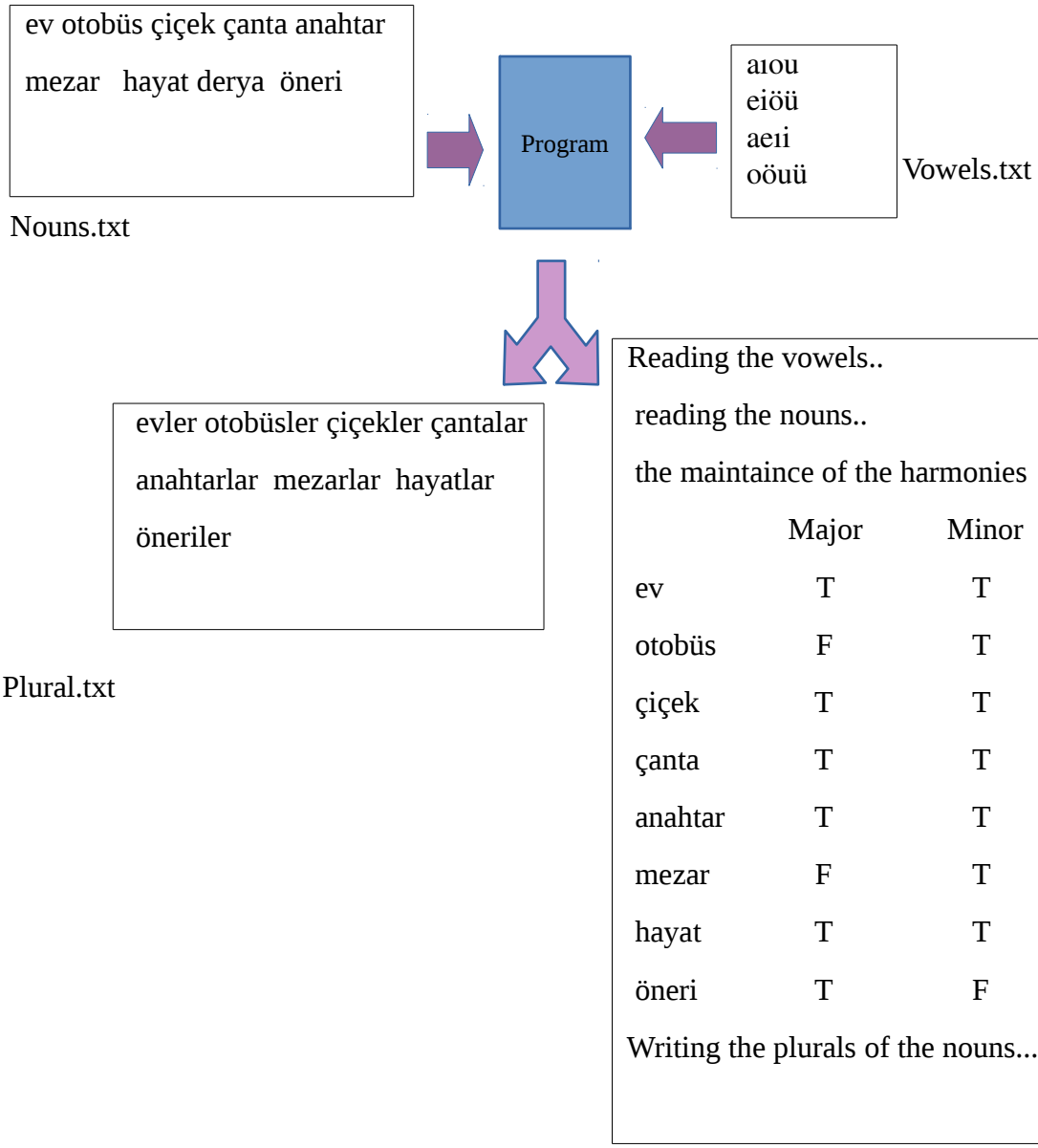
– **major_type find_last_type(const char* word)** returns the major type (HARD or SOFT) of the last vowel. Use the function major in the implementation of this function.

– **char*** **make_plural(const char* noun , char* plural_noun)** takes a string "**noun"** and returns its plural form in "**plural_noun"** (an output argument). It returns a pointer of the output argument as the return value, as well. This function makes the plural noun in accordance with the major vowel rules.

You will use these functions to write the program in main :

1. Read the vowels from the file "**Vowels.txt**"
2. Read the nouns from the "**Nouns.txt**"
3. Print the results of the nouns whether they satisfy the major or minor vowels harmony
4. Write the plural of the nouns into a file called "**Plural.txt**"

Sample files and terminal window:



General:
1.Obey honor code principles.
2.Obey coding convention.
3.Read your homework carefully and follow the directives about the I/O format (data file names, file formats, etc.) and submission format strictly. Violating any of these directives will be penalized.
4.Your submission should include the following file and NOTHING MORE (no data files, object files, etc):

HW06_<student_name>_<studentSirname>_<student number>_part1.c
HW06_<student_name>_<studentSirname>_<student number>_part2.c

5.Do not use non-English characters in any part of your homework (in body, **file name**, etc.).
6.Deliver the printout of your work **until the last submission date**.