ECE 175: Computer Programming for Engineering Applications

Homework Assignment 6

Due Date: Tuesday October 17, 2017 by 11:59 PM, via D2L Dropbox and Zylab

Conventions: Name your C programs as *hw***x***p***y***.c* where *x* corresponds to the homework number and *y* corresponds to the problem number such as hw6p1.c for problem 1 of homework 6.

Write comments to your programs. Three points deduction for each *program with no comment.*
For each program that you turn in, at least the following information should be included at the top of the C file:

- Author:
- Date created:
- Brief description of the program:
    - input(s):
    - output(s):
    - brief description or relationship between inputs and outputs

Submission Instructions: Submit your .c files (hw6p1.c, …) in D2L Dropbox and Zylabs
Make sure that your user-defined functions are named EXACTLY as given in the handout. Otherwise, it will be incorrect on Zylab.

Problem 1 (30 points): IMEI (International Mobile Equipment Identifier)
IMEI is a **unique number of 15 digits** given to every mobile phone.
resource: https://en.wikipedia.org/wiki/International_Mobile_Equipment_Identity

A method to check if the device is really made by the official manufacturer is to compare the ***IMEI's last digit***, called ***Luhn digit***, with a check digit. If the *Luhn digit* is equal to the *check digit*, the device is most probably authentic. Otherwise, it is not authentic for sure.

The check digit is calculated as follows:

1) First, we calculate the **SUM** of the **first IMEI's 14 digits** by **adding**
    a) the **digits in the odd** positions
    b) the ***double of*** the **digits in the even** positions. If the double of the digit is a two-digit number, we add each digit separately. For example, suppose that the value of the digit is 8, its double is 16. We therefore add to the **SUM** the result of 1+6 = 7 (and not 16).
2) If the last digit of the calculated SUM is 0, that is the check digit. If not, we subtract the last digit of the calculated SUM from 10 and that is the check digit.

For example: let's check the IMEI 35768303625737**8**.
Note: the last digit of the given IMEI above is **8** which is the Luhn digit.
Apply the above algorithm (by hand) to the ***first 14 digits***, we get

⇨ 3 + (2x5) +7 + (2x6) + 8 + (2x3) + 0 + (2x3) + 6 + (2x2) + 5+ (2x7) + 3+ (2x7)

⇨ 3 + (10) +7 + (12) + 8 + (6) + 0 + (6) + 6 + (4) + 5+ (14) + 3+ (14)

⇨ 3 + (1+0) +7 + (1+2) + 8 + (6) + 0 + (6) + 6 + (4) + 5+ (1+4) + 3+ (1+4)

⇨ 3 + (1) +7 + (3) + 8 + (6) + 0 + (6) + 6 + (4) + 5+ (5) + 3+ (5)

= 62

⇨ Then the check digit = 10 – 2 = 8.

Since the check digit (=8) is equal to the Luhn digit (=**8**), this IMEI is valid.

Write a **C program** that

- Read the list of IMEI from the “List_IMEI.txt” file.
- For each IMEI value (a 15-element array with an IMEI value),
  - Display IMEI value using the following function:

    ```
    void print_IMEI(int IMEI[])
    {
    //display 15 digit of IMEI value
    }
    ```

  - Calculate and display the sum of the first 14 digits

    Your program must use the following function:

    ```
    void calc_sum(int IMEI[], int *sum_14digits)
    {
            //input: an array of 15 elements
            //output: a total sum of 14 digits of the IMEI array using the
            //algorithm on page 1
    }
    ```

  - Calculate and display the check digit
  - Display the Luhn digit
  - Compare the check digit with the Luhn digit.
    - If they match, display the message “IMEI is valid.”
    - Otherwise, display with the message “IMEI is invalid”.

Suggestion: the fscanf function reads one integer at a time. Keep each integer into the 15-element array. Every 15 integers read in from the text file, it is one complete IMEI value.

Sample code execution: **For the FIRST 2 IMEI values in the text file**

```
IMEI: 357683036257378
total sum of 14 digits = 62
Luhn digit = 8
check digit = 8
IMEI is valid

IMEI: 357683036257379
total sum of 14 digits = 62
Luhn digit = 9
check digit = 8
IMEI is invalid
```

Problem 2 (40 points): Write an interactive C program that lets a user play a game of **Hangman**

1) Your program gets the secret word (7 letters in each word) by calling the given function:

```
void get_word(char word[]) {
      char WORD[][8] = { "program", "puzzles", "squeeze",
                         "circuit", "devoted", "journey",
                         "version", "totally", "respect" };
      int i, num;
      num = (rand() % 9);
      for (i = 0; i < 7; i++)  //exclude NULL
            word[i] = WORD[num][i];
}
```

Suggestion: During the time that you implement your code, pick one word to test your code and hard-code it in your main program, i.e. `char word[7] = "squeeze";` After your code works, call the `get_word` function to randomly pick a secret word.

2) In your main program, include the followings (so that the rand() function in `get_word` works correctly)

```
#include <stdlib.h>   // enable use of rand()
#include <time.h>     // enable use of time()
```

and

```
srand((int)time(0));
```

3) Initially the program prints on the screen the *number of letters* of the ***word*** to be guessed. This is in the form of successive stars (see sample code execution on the next page).

The player makes a guess on the letters belonging to the secret ***word*** one by one. At each step, the program prints on the screen the letters that have been guessed, and the number of wrong guesses.

The program should terminate when either

a) all letters have been guessed correctly (the player wins) or
b) a specified number of incorrect guesses have been made (the computer wins).

Your program must include the following function

```
void print_word(char x[], int len)
{
      //your code here to print elements of array x
      //(there are len elements in array x)
}
```

Suggestion: Use another array, guessed, to keep track of the solution so far. Initialize all elements of guessed to the '*' symbol. Each time a letter in ***word*** is guessed correctly, replace the corresponding '*' in guessed with that letter.

Sample code execution for word ***squeeze***

```
Let's play Hangman. The secret word is:
*******

Guess a letter: e
e was found 3 times in the secret word

***ee*e
Guess a letter: t
t is not in the secret word, You have 6 tries left.

***ee*e
Guess a letter: z
z was found 1 times in the secret word

***eeze
Guess a letter: a
a is not in the secret word, You have 5 tries left.

***eeze
Guess a letter: q
q was found 1 times in the secret word

*q*eeze
Guess a letter: s
s was found 1 times in the secret word

sq*eeze
Guess a letter: y
y is not in the secret word, You have 4 tries left.

sq*eeze
Guess a letter: u
u was found 1 times in the secret word

squeeze
Congratulations! You found the secret word: squeeze
```

```
Let's play Hangman. The secret word is:
*******

Guess a letter: a
a is not in the secret word, You have 6 tries left.

*******
Guess a letter: t
t is not in the secret word, You have 5 tries left.

*******
Guess a letter: s
s was found 1 times in the secret word

s******
Guess a letter: u
u was found 1 times in the secret word

s*u****
Guess a letter: x
x is not in the secret word, You have 4 tries left.

s*u****
Guess a letter: o
o is not in the secret word, You have 3 tries left.

s*u****
Guess a letter: m
m is not in the secret word, You have 2 tries left.

s*u****
Guess a letter: w
w is not in the secret word, You have 1 tries left.

s*u****
Guess a letter: k
k is not in the secret word, You have 0 tries left.

s*u****
Game over! The secret word was: squeeze
```