Department of Computer Engineering

CENG104 – Computer Programming II

Spring 2017 - 2018

Lab Guide #2/C – Week 3

OBJECTIVE : Structure and Pointer Operations

Instructor : Yusuf Evren AYKAÇ
Assistants : Elif GÜL, Yusuf Şevki GÜNAYDIN, Hatice ÇATALOLUK

Q1.

Create the structure **product_t** with the following fields.

```
typedef struct{
        char prod_name[20];
        double  price;
        int warranty;
}product_t;
```

Write a modular C program that will read the information of the products from the **product.txt** file, store them into an array of structure, and display the number of products and the information of all products. The first digit at the very beginning of the text file denotes the count of the products. Don't forget to use pointer notation.
(Hint: ***use %s operator while reading or displaying the product name***.)
Before you start;
Lexicon*: Warranty → type of guarantee*

Example Run:

```
There are 3 products

The Product Information:
************************
Name : Arcelik
Price: 251.0
Years of Warranty: 3
************************
Name : Vestel
Price: 355.99
Years of Warranty: 2
************************
Name : Beko
Price: 410.59
Years of Warranty: 5
```

Product.txt

```
3
Arcelik 251.0 3
Vestel 355.99 2
Beko 410.59 5
```

Project Name: LG2_AQ1
Source Name: AQ1.cpp

Q2.

Numeric addresses for computers on the Internet are **unique** and composed of four parts, seperated of four parts, seperated by periods of the form **xxx.yyy.zzz.www** where xxx, yyy, zzz, and www are positive integers. Locally computers are usually known by nickname as well.

You are designing a program to process a list of Internet addresses, identifying all pairs of computers from the same locality. Create a structure type called **address_t** with components for the four integers of an Internet address and fifth component in which to store an associated nickname of a character.

Your program should be capable of

- reading a list of up to 100 addresses and nicknames from the user until a sentinel address of all zeros’ is entered.
- controlling if a specified nickname already exists in the list or not. (**Hint**: Write a function **search_nick** that checks and returns 1 if a given nickname occurs before, 0 otherwise.) Program should ask for a nickname until a non-existing nickname is entered. (Examine example runs carefully!)
- displaying a list of messages identifying each pair of computers from the same locality - that is, each pair of computers with matching values in the first two components of the address. In the messages, computers should be identified by their nicknames and Internet addresses. If no computers are found from the same locality, display a warning message also.

Example Run#1:

```
Numeric addresses are specified in the form of xxx.yyy.zzz.www

Enter the numeric address and nickname: 145.65.0.21 T
Enter the numeric address and nickname: 126.84.2.11 A
Enter the numeric address and nickname: 157.22.88.65 a
Existing nickname!
Enter another nickname: A
Existing nickname!
Enter another nickname: G

Enter the numeric address and nickname: 145.65.24.10 F
Enter the numeric address and nickname: 157.22.14.3 H
Enter the numeric address and nickname: 0.0.0.0 m

Computers
'T' (145.65.0.21)
'F' (145.65.24.10)
'G' (157.22.88.65)
'H' (157.22.14.3)
are from the same locality!
```

Example Run#2:

```
Numeric addresses are specified in the form of xxx.yyy.zzz.www

Enter the numeric address and nickname: 156.21.78.1 S
Enter the numeric address and nickname: 187.32.1.21 b
Enter the numeric address and nickname: 165.200.0.41 C
Enter the numeric address and nickname: 255.214.213.1 F
Enter the numeric address and nickname: 127.0.0.1 f
Existing nickname!
Enter another nickname: L

Enter the numeric address and nickname: 0.0.0.0 h

No computers are from the same locality!
```

Project Name: LG2_AQ2
File Name: AQ2.cpp