CMPE – 211

Preliminary Work (Pre-Lab Activity)

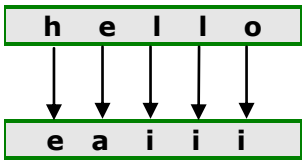
Laboratory Experiment #4

Textbook Material:	Chapters 5- 7	*pp.87-182*	[see Laboratory Experiment #3]
	Chapter 8	*«C-Strings»*	*pp.183-196*
	Chapter 9	*«Standard C++ Strings»*	*pp.213-230*

• • • TASK 1

Write, Compile and Execute a C++ program that **converts a string** into another string as follows: Each character of the original string is replaced by the closest vowel placed before that character - for example, (b,c,d) are replaced by a, (f, g, h) are replaced by e and e is replaced by a, where a remains the same. You are expected to use "plain" arrays of characters for the string representation, i.e. so-called C-strings [→ C++ supports them together with a standard type for strings called **`string`** – do You think that a given problem can be solved using standard C++ strings; if yes, write a variant of the problem that utilizes C++ strings instead of C-strings].

Example: string **"hello"**, which consists of lowercase letters only, is converted to the string **"eaiii"**. As a result, conversion on the character-by-character basis can be depicted as follows:



- **NOTE:** For this and all following tasks You are free to use any basic constructions and data types (e.g. *scalar*, i.e. integral, floating-point, pointers; *compound*, e.g. enumerations, arrays, etc.) of C++ the most appropriate for the implementation.

• • • TASK 2

Write, Compile and Execute a C++ program that checks whether a given substring is found in a given string. If found, it is removed from the string. Please, check carefully the list standard C-string functions (header **`<string>`**, pages 199-200 in the textbook).

Example: Consider a string **"tank capacity was measured many times"** to search a substring **"an"** specified by a user. Hypothetical dialogue may have the form given below:

```
Input string for search:  tank capacity was measured many times
Input substring to search for:  an
After removing: tk capacity was measured my times
```

• • • TASK 3

Write, Compile and Execute a C++ program that inputs a telephone number as a string in the form **(392) 630-1234**. The program should use standard function **`strtok( )`** (see pages 199-200 in the textbook) to extract the area code as a token, the first three digits of the phone number as a token, and the last four digits of the phone number as a token. The seven digits of the phone number should be concatenated into one string. Both the **area code** and the **phone number** should be **printed**.

The initial part of the program may look as follows (You are free to follow it or not):

- continued next page -

```
#include <iostream>
using namespace std;
...  ...  <other files for inclusion> ...  ...
#include <cstring>

int main( )
{
        const int SIZE1 = 20;
        ...  ...  <other declarations, if needed> ...  ...
        char p[SIZE1];
```

```
        char *tokenPtr;        // store temporary token
        char *areaCode;        // store area code
        char *phone;           // store the phone number
        cout << "Enter a phone number in the form (555) 555-5555:  ";
        cin.getline(p, SIZE1);   ...  ...  ...
}
```

Hypothetical dialogue may have the form given below:

```
Enter a phone number in the form (555) 555-5555:  (392) 630-1234
Output:
Area code (token 1) is 392
Telephone number is 6301234
```

● ● ● TASK 4

Fill in the blanks in each of the following:

1. Header _____ must be included for class **string**,
2. Class **string** belongs to the _____ **namespace**,
3. Concatenation of **string** objects can be performed with operator _____ ,
4. (State *true* or *false*) A C-style string is a **string** – it is _____ .

Write, Compile and Execute a C++ program that separately inputs two names (objects of the class **string**) and inserts the shorter into the longer to create a new **string**. The output should have the following form:

```
Specify the first name:  Peter
Specify the second name:  Grunefeld
The replaced form is Peterfeld

Specify the first name: abcdng
Specify the last name:  cmpe
The replaced form is cmpeng
```

● ● ● Appendix

- Check Review Questions at the end of textbook's Chapters 7, 8 and 9 (pp. *174*, *195* and *221*) and review pointer arithmetic rules, C-string processing and standard C++ Strings,
- Use debugging facilities of the Visual C++ compiler while writing programs for TASKS 1-4 (follow tutorial «Dive Into Microsoft Visual C++ 6» by Deitel & Associates)

● ● ● Sources

- *John R. Hubbard*. Schaum's Outline of Programming with C++, 2nd edition, McGraw-Hill, 422 p., 2000
- *Harvey M. Deitel, Paul J. Deitel*. C++ How To Program, 4th edition, Prentice Hall, 1320 p., 2002
- Strings (Lesson 9), http://www.cprogramming.com/tutorial/lesson9.html
- C/C++ Reference (N.Kohl, 2005) – C++ Strings, http://www.cppreference.com/cppstring/
- C++ String class Examples and Tutorial (G.Ippolito, 2001-2003), http://www.yolinux.com/TUTORIALS/LinuxTutorialC++StringClass.html