

## Lab 12

### Show the homework tasks first

[Strings handling] Write a C program for solving **Programming questions 8.14 to 8.16 and 8.23 to 8.29** on **Pages 429** of your textbook (Deitel, 9<sup>th</sup> Edition).

**8.14** (*Tokenizing Telephone Numbers*) Write a program that inputs a telephone number as a string in the form (555) 555-5555. Use function `strtok` to extract as tokens the area code, the first three digits of the phone number and the last four digits of the phone number. Concatenate the phone number's seven digits into one string. Convert the area-code string and phone-number string to integers, then display both.

**8.15** (*Displaying a Sentence with Its Words Reversed*) Write a program that inputs a line of text, tokenizes the line with function `strtok` and outputs the tokens in reverse order.

---

**8.23** (*Strings Starting with "b"*) Write a program that reads a series of strings and prints only those beginning with the letter "b".

**8.24** (*Strings Ending with "ed"*) Write a program that reads a series of strings and prints only those that end with the letters "ed".

**8.25** (*Printing Letters for Various ASCII Codes*) Write a program that inputs an ASCII code and prints the corresponding character.

**8.26** (*Write Your Own Character-Handling Functions*) Using the ASCII character chart in Appendix B as a guide, write your own versions of the character-handling functions in Section 8.3.

**8.27** (*Write Your String-Conversion Functions*) Write your own versions of the functions in Section 8.4 for converting strings to numbers.

**8.28** (*Write Your Own String-Copy and String-Concatenation Functions*) Write two versions of each string-copy and string-concatenation function in Section 8.6. The first version should use array indexing, and the second should use pointers and pointer arithmetic.

**8.29** (*Write Your Own String-Comparison Functions*) Write two versions of each string-comparison function in Fig. 8.13. The first version should use array indexing, and the second should use pointers and pointer arithmetic.

**8.30** (*Write Your Own String-Length Function*) Write two versions of function `strlen` in Fig. 8.27. The first version should use array indexing, and the second should use pointers and pointer arithmetic.

**8.16** (*Searching for Substrings*) Write a program that inputs a line of text and a search string from the keyboard. Using function `strstr`, locate the first occurrence of the search string in the line of text. Assign the location to variable `searchPtr` of type `char *`. If the search string is found, print the remainder of the line of text beginning with the search string. Then, use `strstr` again to locate the next occurrence of the search string in the line of text. If a second occurrence is found, print the remainder of the line of text beginning with the second occurrence. [*Hint: The second call to `strstr` should contain `searchPtr + 1` as its first argument.*]