12. Modify Programming Project 14 from Chapter 8 so that it stores the words in a two-dimensional `char` array as it reads the sentence, with each row of the array storing a single word. Assume that the sentence contains no more than 30 words and no word is more than 20 characters long. Be sure to store a null character at the end of each word so that it can be treated as a string.

13. Modify Programming Project 15 from Chapter 8 so that it includes the following function:

    ```
    void encrypt(char *message, int shift);
    ```

    The function expects message to point to a string containing the message to be encrypted; shift represents the amount by which each letter in the message is to be shifted.

14. Modify Programming Project 16 from Chapter 8 so that it includes the following function:

    ```
    bool are_anagrams(const char *word1, const char *word2);
    ```

    The function returns true if the strings pointed to by word1 and word2 are anagrams.

15. Modify Programming Project 6 from Chapter 10 so that it includes the following function:

    ```
    int evaluate_RPN_expression(const char *expression);
    ```

    The function returns the value of the RPN expression pointed to by expression.

16. Modify Programming Project 1 from Chapter 12 so that it includes the following function:

    ```
    void reverse(char *message);
    ```

    The function reverses the string pointed to by message. *Hint:* Use two pointers, one initially pointing to the first character of the string and the other initially pointing to the last character. Have the function reverse these characters and then move the pointers toward each other, repeating the process until the pointers meet.

17. Modify Programming Project 2 from Chapter 12 so that it includes the following function:

    ```
    bool is_palindrome(const char *message);
    ```

    The function returns true if the string pointed to by message is a palindrome.

18. Write a program that accepts a date from the user in the form *mm/dd/yyyy* and then displays it in the form *month dd, yyyy*, where *month* is the name of the month:

    ```
    Enter a date (mm/dd/yyyy): 2/17/2011
    You entered the date February 17, 2011
    ```

    Store the month names in an array that contains pointers to strings.