**6.2 STRING PROCESSING EXAMPLES**

In this section, we will study several examples that illustrate the use of string processing functions described in the previous section.

**6.2.1 Counting the Number of Matching Characters in a Pair of Strings**

Given two strings str1 and str2 of alphabets, we wish to find the count of characters in str1 that match a character in str2, ignoring any difference in case (lowercase or uppercase). If a character, say, ch1, in str1 appears more often than once in str2, every occurrence of ch1 in str2 should be counted. For this purpose, we develop the function nMatchedChar (Fig. 6.2).



As we do not wish to distinguish among alphabets based on the case (lower or upper), we convert the input arguments to lowercase. To keep a count of the matched characters, we initialize the variable count to 0. The outer loop works by picking up a character ch1 from temp1 and comparing it against every character ch2 in temp2 in the nested loop. For each matched pair, the variable count is incremented by one. At the end of the function, the value of count is returned. Let us examine an example:

> >> name1 = 'Ram Rahim'

>>> name2 = 'SAMARTH RAHI'

>>> nMatchedChar(name1, name2)

16

Note that the first 'R' in string 'Ram Rahim' matches 'R' in 'SAMARTH RAHI' at indices 4 and 8, first 'a' in string 'Ram Rahim' matches 'A' in 'SAMARTH RAHI' at indices 1, 3, and 9, first 'm' in string 'Ram Rahim' matches 'M' in 'SAMARTH RAHI' at index 2, first space character ' ' in string 'Ram Rahim' matches ' ' in 'SAMARTH RAHI' in at index 7, the second 'R' in string 'Ram Rahim' again matches 'R' in 'SAMARTH RAHI' at indices 4 and 8, and so on.

**6.2.3 Reversing a String**

Next, we wish to find the reverse of a given string. For example, on reversing the string 'abcddb' we obtain the string 'bddcba'. To find reversed string of a given string str1, we begin with an empty string reverseStr. In case, the input string is a null string, the reversed string reverseStr will also be a null string. In the case of a non-null string, for each character in the given string, we concatenate by appending it with the reverseStr built so far. The complete function reverse is given in Fig. 6.4.

