Practice questions v5: Char arrays strings

1. **Which of the following declarations are equivalent?**
   ```
   char string_var[10] = "Hello";
   char string_var[10] = {'H', 'e', 'l', 'l', 'o', '\0'};
   char string_var[10] = {'H', 'e', 'l', 'l', 'o'};
   char string_var[6] = "Hello";
   char string_var[] = "Hello";
   ```

2. **Explore strcat function and answer the following:**
   a. What string will be stored in singing_string after the following code is run?
   ```
   char singing_string[20] = "DoBeDo";
   strcat(singing_string, " to you");
   ```
   b. What string will be output when this code is run? (Assume, as always, that this code is embedded in a complete, correct program.)
   ```
   char song[10] = "I did it ";
   char franks_song[20];
   strcpy( franks_song, song );
   strcat( franks_song, "my way!");
   cout << franks_song << endl;
   ```
   c. What is the problem (if any) with this code?
   ```
   char a_string[20] = "How are you? ";
   strcat(a_string, "Good, I hope.");
   ```

3. **What is the maximum length of a string that can be placed in the string variable declared by the following declaration? Explain.**
   ```
   char s[6];
   ```

4. **How many characters are in each of the following character and string constants?**
   a. '\n'
   b. 'n'
   c. "Mary"
   d. "M"
   e. "Mary\n"

5. **Given the following declaration and initialization of the string variable, write a loop to assign 'X' to all positions of this string variable, keeping the length the same.**
   ```
   char our_string[15] = "Hi there!";
   ```

6. **Consider the following code (and assume it is embedded in a complete and correct program and then run):**
   ```
   char my_string[80];
   cout << "Enter a line of input:\n";
   cin.getline(my_string, 6);
   cout << my_string << "<END OF OUTPUT";
   ```
   If the dialogue begins as follows, what will be the next line of output?
   ```
   Enter a line of input:
   May the hair on your toes grow long and curly.
   ```

7. **Consider the following code:**
   ```
   string s1, s2("Hello");
   cout << "Enter a line of input:\n";
   cin >> s1;
   if (s1 == s2)
   cout << "Equal\n";
   else
   cout << "Not equal\n";
   ```
   If the dialogue begins as follows, what will be the next line of output?
   ```
   Enter a line of input:
   Hello friend!
   ```

8. **What is the output produced by the following code?**

```
string s1, s2("Hello");
s1 = s2;
s2[0] = 'J';
cout << s1 << " " << s2;
```

9. **Create a Cstring variable that contains a name, age, and title. Each field is separated by a space. For example, the string might contain "Bob 45 Programmer" or any other name/age/title in the same format. Assume the name, age, and title have no spaces themselves. Write a program using only functions from cstring (not the class string) that can extract the name, age, and title into separate variables. Test your program with a variety of names, ages, and titles.**

10. **Write a program that inputs a first and last name, separated by a space, into a string variable. Use the string functions to output the first and last initial. Embed your code into a *do-while* loop. At the end of the loop ask the user if he or she would like to repeat the program. Input the user's choice into a char using cin. If the character is 'y' then repeat the program, otherwise exit. Beware of the pitfall with newlines when cin is mixed with getline.**

11. **Write a program that will read in a line of text and output the number of words in the line and the number of occurrences of each letter. Define a word to be any string of letters that is delimited at each end by either whitespace, a period, a comma, or the beginning or end of the line. You can assume that the input consists entirely of letters, whitespace, commas, and periods. When outputting the number of letters that occur in a line, be sure to count upper- and lowercase versions of a letter as the same letter. Output the letters in alphabetical order and list only those letters that do occur in the input line. For example, the input line**

```
I say Hi.
```

should produce output similar to the following:

```
3 words
1 a
1 h
2 i
1 s
1 y
```

12. **Write a program that reads in a sentence of up to 100 characters and outputs the sentence with spacing corrected and with letters corrected for capitalization. In other words, in the output sentence, all strings of two or more blanks should be compressed to a single blank. The sentence should start with an uppercase letter but should contain no other uppercase letters. Do not worry about proper names; if their first letters are changed to lowercase, that is acceptable. Treat a line break as if it were a blank, in the sense that a line break and any number of blanks are compressed to a single blank. Assume that the sentence ends with a period and contains no other periods. For example, the input**

```
the Answer to life, the Universe, and everything
IS 42.
```

should produce the following output:

```
The answer to life, the universe, and everything is 42.
```

13. **Write a program that reads in a line of text and replaces all four-letter words with the word "love". For example, the input string**

```
I hate you, you dodo!
```

should produce the output

```
I love you, you love!
```

Of course, the output will not always make sense. For example, the input string

```
John will run home.
```

should produce the output

```
Love love run love.
```

If the four-letter word starts with a capital letter, it should be replaced by "Love", not by "love". You need not check capitalization, except for the first letter of a word. A word is any string consisting of the letters of the alphabet and delimited at each end by a blank, the end of the line, or any other character that is not a letter. Your program should repeat this action until the user says to quit.

14. **Write a program that reads in a line of text and outputs the line with all the digits in all integer numbers replaced with 'x'. For example**,
    Input:
    ```
    My userID is john17 and my 4 digit pin is 1234 which is secret.
    ```

    Output:
    ```
    My userID is john17 and my x digit pin is xxxx which is secret.
    ```

    Note that if a digit is part of a word, then the digit is not changed to an 'x'. For example, note that john17 is NOT changed to johnxx. Include a loop that allows the user to repeat this calculation again until the user says she or he wants to end the program.