Exercise – Character Arrays

Exercises:

1. Create a program that asks for the user's first and last name and store it in one string. It must be able to handle spaces in the input.
2. Output the name in reverse order
3. Modify the program to take in 5 names and print them out in reverse order too. You will need to us a 2D character array for this.
4. Research and find out what character code causes your computer to beep
5. Create a program that allows the user to enter a username and password. Ensure the password is at least 8 characters long and contains at least one capital letter, at least one number and at least one symbol.
6. Create the following program that demonstrates the use of strlen, strcmp, strcat and strcpy:
    a. Ask the user to enter a username. Only let the program continue if the username contains the sequence 'usr' within it.
    b. Ask the user to enter a password. Only let the program continue if the password is exactly the same as 'Pa55w0rd'
    c. Construct a new char array that contains the user's identifier. The identifier will consist of the username, followed by a '-', and the password. For example, if the username/password was 'usrJohn', 'Pa55w0rd' respectively, the identifier will be 'usrJohn-Pa55w0rd'. Output this identifier to the console.
7. Each of the following questions shows code that has problems. For each them, answer the following:
    - What do you think the program is **trying** to achieve?
    - What is the bug?
    - How would you correct it?

    a.

```
char name[4] = "John";
```

    b.

```
char name[32];
cout << name;
```

c.

```
char name[7] = "Donald";
char surname[7] = "Knuth";

strcat(name, surname);
```

d.

```
char errorMsg[5] = "Stop";
errorMsg[strlen(errorMsg)] = "!";

cout << errorMsg;
```

8. Write a program that takes in a word from the user and outputs it in pig latin.

9. Write a program that takes a line of input and counts the number of words and letters.

10. Write a program that takes a word as input from the user and prints out whether or not it's a palindrome.

11. Implement your own versions of strlen, strcmp, strcat, and strcpy.

12. Write a program that reads in a line of input and then displays the character that appears the most frequently in that sentence.

13. Write a program that reads in a date from the user in the form dd/mm/yyyy. The program should then print out the date in the form: March 12, 2014

14. Given a line of input as a string, create a copy of the string that has all whitespace removed.

15. Create a function that encodes a string by offsetting each character by 5 values. For example 'a' should become 'f'. The function should return the encoded string. Do **not** pass in the string length to the function – work it out using string functionality.