Assignment 9
Class Exercise, worth 5 points.

Arrays of characters are common in C++. This exercise is about using an array of characters. Create a main function and within main declare a large array of char's as follows:

```
char cBuffer[256];
```

Actually, the more correct way to do that is as follows:

```
const int BUF_SZ = 256;
char cBuffer[BUF_SZ];
```

Now use that array with the following 4 functions.

[Note that C++ can treat character arrays much like a string, and not just an array. For example, you can do either of the following, cout << cBuffer; and cin >> cBuffer;. You may use these two statements to test your code. But for this assignment you must read or write one character at a time in a loop, not a whole array.]

[Also note that cin and cout are provided by the C++ language to make it easier to debug programs. They are not designed for production use. There are situations where they do strange things. Don't spend time trying to make them act perfectly.]

1. Write a function called getWord() that takes and array of char and an int which is the length of the array. Inside the function, prompt the user to enter a word ending with a comma. Then read the letters (chars) that the user has typed at the console, and put them in the array. For this function you must use a loop and read the characters into the array one character at a time. If you reach the end of the array (the second parameter passed to this function), stop reading chars. Otherwise, simply stop when you read the comma. At that point, replace the comma with the value 0. The function should return the length of the word that was read. (In the case that you somehow reach the end of the buffer, return that length instead.)

2. Write a function called showWord() that takes an array of char and the number of chars to print. (This length is the length of the word, not the length of the buffer.) Inside the function, print the word by looping and printing one char at a time. The output should look like a word – all on the same line. This function does not return anything.

3. Write a function to count the number of times the letter 'e' occurs in the word.

4. Write a function that takes an array of char and replaces any lower case letter with its upper case equivalent. (Note that you can do integer operations on char type variables. In this case, you want to test to see if it is in the range of lower case letters, between 'a' and 'z'. If it is, you want to add to the value that is the difference between ASCII 'A' and ASCII 'a'.)

Include code in main() to exercise each of these functions.

Turn in your cpp file when you have all four functions working.