// enable array initialization

char *wail = "ululate";    // wail points to string

int ms = c_in_str(mmm, 'm');
int us = c_in_str(wail, 'u');
cout << ms << " m characters in " << mmm << "\n";
cout << us << " u characters in " << wail << "\n";
return 0;
}

// this function counts the number of ch characters
// in the string str
int c_in_str(const char * str, char ch)
{
    int count = 0;

    while (*str)        // quit when *str is '\0'
    {
        if (*str == ch)
            count++;
        str++;        // move pointer to next char
    }        // move pointer to next char
    return count;
}

Here's the output:

3 m characters in minimum
2 u characters in ululate

Program Notes

Because the c_in_str() function shouldn't alter the original string, it uses the const modifier when it declares the formal parameter str. Then, if you mistakenly let the function alter part of the string, the compiler catches your error. Of course, you can use array