# CS 2275 Exercise 3 – Strings

Module 3 objectives: C++ strings and string gadgets  
  
Readings for Module 3: Lippman 3.1-3.4,8, 9.5

Using Dev-C++, create a .cpp source file titled ex3-<your last name>.cpp that contains solutions to the following problems. Use a .doc or .txt file ex3<your last name>.doc to provide the answers to any text questions. As with all assignments in this course, be sure to include a block comment just about the function indicating its purpose and use a really good function name. Include code under main to test your function by allowing the user to enter the needed parameters. Make sure you test your function sufficiently to insure correctness.

1. Write a c++ function countChar(char c, const string &s) that accepts a character and a string and returns the number of occurrences of c in s. Thus countchar( ‘c’, “abcdfecc”) would return 3.
   1. What was your test plan for this function that you used to determine it worked correctly?
2. Write a c++ function reverse (const &s) that accepts a string and returns its reverse. Thus reverse (“abcde”) would return “edcba”.
3. Write a c++ function that accepts a string and returns the sum of all the asci values of all the characters in that string.
4. Write a c++ function parceHex(const string &hexString) that accepts a positive hexnumber as a string and returns the corresponding decimal integer. Thus parceHex (“FD”) would return 254. There is a built in gadget that does this, but don’t use it. Rather generate your own gadget using basic code.
5. Write a C++ function that accepts a string and changes all sequences of blanks into single blanks. Use the provided forceSingleSpaces as a starting place, but develop a more elegant solution that does not use a Boolean variable and has significantly fewer lines of code.
6. Write 4 C++ functions:
   1. Write a C++ function ROT128char that accepts an asci character, adds 128 to it asci value, applies mod 256, and returns the resulting character.
   2. Write a function that accepts a ROT128 character and reverses it back to the original. (Think carefully about what is needed for this second function.)
   3. Write a function ROT128 that accepts a string s and returns a string containing the ROT128 versions of all the characters in S in their original order.
   4. Write a function that generates the original s from the string that has been rotated by ROT128.
   5. What tests are needed to test all four of your functions?

Submit your .cpp and .doc files using moodle

Grading rubric: Problem 1 20 points. Problems 2 through 5: 10 points. Problem 6 30 points. Style – 10: poor variable or poor function names or no block comment with a goal statement for a function. Solutions that have significantly more lines that needed will be docked points for lack of elegance.

**REMINDERS**

* code that does not compile will not be graded
* the grader should not need to modify/uncomment your code to test it. Provide a test mechanism allowing the grader to enter various tests for teach function.