Find the First Non-Repeated Character

Write an efficient function to find the first non-repeated character in a string. For instance, the first nonrepeated character in "total" is 'o' and the first non-repeated character in "teeter" is 'r'. Discuss the efficiency of your algorithm.

Improved Unicode+Efficient First Non-Repeated Character

Same as the previous problem but now implement 2 improvements: support for Unicode code points and remove the inefficiency involved with incrementing an Integer object each time.

Remove Specified Characters

Write an efficient function that deletes characters from an ASCII string. string removeChars(string str, string remove); Any character existing in "remove" must be deleted from "str". "Battle of the Vowels: Hawaii vs. Grozny" and remove of "aeiou" would return "Bttl f the Vwls: Hw vs. Grzny".

Remove Specified Characters with Array(s)

Same as the previous problem but now implement with only array(s).

Reverse Words

Write a function that reverses the order of the words in a string. For example, your function should transfrom the string "Do or do not, there is no try." to "try. no is there not, do or Do". Assume that all words are space delimited and treat punctuation the same as letters.

Integer/String Conversions

Write 2 conversion routines. The first routine converts a string to a signed integer. You may assume that the string contains only digits and the minus character ('-'), that is a properly formatted integer number, and that the number is within the range of an int type. The second routine converts a signed integer stored as an int back to a string.