

INFORMATICA GENERALE

Homework 1: Parole, Parole, Parole, ...

docente: IVANO SALVO
Sapienza Università di Roma

pubblicazione: 27.IV.2015 - consegna 10.V.2015

Esercizio 1: (ANAGRAMMI) Scrivere un programma C che legge in input due stringhe s_1 ed s_2 e verifica se s_1 è un anagramma di s_2 . In caso affermativo, stampa 1, altrimenti stampa 0. Attenzione al fatto che eventuali caratteri ripetuti *devono avere lo stesso numero di occorrenze* in s_1 ed s_2 .

ESEMPI: La stringa *roma* è anagramma di *mora*, ma non di *morra*.

Esercizio 2: (CODICI SEGRETI) Scrivere un programma C che legge in input una stringa s ed un intero k e restituisce in output una stringa in cui tutti i caratteri di s sono spostati di k posizioni in avanti. I caratteri dell'alfabeto (inglese) vengono interpretati in modo circolare, per cui il successivo di 'z' è 'a'.

ESEMPI: Data in input la stringa **renzi** e l'intero 3, l'output deve essere **uhqcn**. Se $k = 0, 26, 52, \dots$ la stringa rimane immutata.

Esercizio 3: (FACOLTATIVO – vale un bonus di 1 punto sul voto finale) Scrivere un programma C che legge in input una stringa s e produce in output la prossima permutazione di s nell'ordine lessicografico. Se s è l'ultima permutazione, il programma dovrà stampare 0. L'ordine lessicografico è il comune ordine usato dal dizionario.

ESEMPIO: le permutazioni di *roma* sono nell'ordine lessicografico: amor, amro, aomr, aorm, armo, arom, maor, maro, moar, mora, mrao, mroa, oamr, oarm, omar, omra, oram, orma, ramo, raom, rmao, rmoa, roam, roma.

Quindi, se l'input fosse **oram** l'output deve essere **orma**. Se l'input fosse **roma**, l'output deve essere 0.