

## Problemi:

1. Scrivere un programma `trova_parola` `int(char *input, char *target)` che accetta in `input` due stringhe, `input` e `target` e restituisce il numero di volta che `target` apparisce in `input`: per esempio, data "the quick brown fox drowned" come `input` e "own", la funzione restituisce 2.
2. Scrivere un programma che prenda in `input` due linee di testo contenenti ciascuna un numero binario (rappresentato da una sequenza di cifre binarie seguite da `\n`), immagazzinando ciascuno di essi una stringhe, i cui elementi rappresentano singole cifre binarie.

Si scriva poi una funzione `somma_numeri_binari` che effettua la somma di due numeri binari rappresentati come stringhe, restituendo il risultato in una terza stringhe.

- 3. Scrivere una funzione** `void summatrix(int n, int A[n][n], int B[n][n])` che prese in input due matrici  $n \times n$  di interi modifica la matrice `A` rendendola uguale alla somma delle due matrici (elemento per elemento). Ad esempio, se  $n = 3$  e

$$\begin{array}{r} \phantom{A = } 2 \ 4 \ 1 \\ A = 3 \ 5 \ 10 \\ \phantom{A = } 2 \ 7 \ 11 \end{array} \qquad \begin{array}{r} \phantom{B = } 2 \ 1 \ 0 \\ B = 5 \ 5 \ 8 \\ \phantom{B = } 2 \ 3 \ 9 \end{array}$$

allora la funzione modifica la matrice `A` così:

$$\begin{array}{r} \phantom{A = } 4 \ 5 \ 1 \\ A = 8 \ 10 \ 18 \\ \phantom{A = } 4 \ 10 \ 20 \end{array}$$

4. Dato un matrice  $A[n][n]$  un elemento  $A[i][j]$  detto "column and row max" se  $A[i][j]$  è maggiore o uguale a tutti i elementi che sono in lo stesso colonna o riga di  $A[i][j]$ . Scrivere una funzione `int colrow_maxsum(int n, int A[n][n])` che accetta in input un intero  $n$  e un  $n \times n$  matrice  $A$  e restuisce la somma di tutti i column and row max elementi. Ad esempio, se  $n = 3$  e

$$A = \begin{matrix} & 8 & 4 & 1 \\ 3 & & 5 & 10 \\ 2 & 7 & 11 & \end{matrix}$$

allora i valori  $A[0][0]$  e  $A[2][2]$  sono column and row max e la funzione restuisce 19.