

Esercizio 1

Scaricare il programma `quick_sort_parziale.c` che si trova sulla pagina del corso. Questo sorgente contiene un'implementazione di Quick Sort che deve essere completata scrivendo il corpo della funzione

```
int distribuzione(int a[], int sx, int px, int dx)
```

Questa funzione deve partizionare gli elementi dell'array $a[sx \dots dx]$ utilizzando l'elemento $a[px]$ come pivot e restituire la posizione di tale elemento dopo il partizionamento. Scrivere un main di test che legga un array di N interi da tastiera (dove la codifica è quella usuale, ovvero il primo intero determina la lunghezza dell'array e poi seguono gli elementi dell'array) e successivamente ordini l'array usando la funzione `quick_sort` ed infine stampi gli elementi ordinati.

NB: usare la funzione `rand()`, per generare un numero casuale e l'operatore modulo `%` per limitare entro un certo intervallo tale numero.

Esercizio 2

A partire dal codice dell'esercizio precedente, implementare il Quick Sort per stringhe. Scrivere quindi un main di test che si comporti allo stesso modo del precedente, ma da input verrà letto un array di stringhe, codificato come al solito.

Esercizio 3

Modificare il Quick Sort dell'esercizio 1, in maniera tale che ordini gli elementi pari nella parte inferiore dell'array e quelli dispari in quella superiore. Scrivere quindi un main di test che legga un array di interi da input come al solito ed esegua la funzione.

Esempio: Dato un array $A = [1, 78, 42, 54, 23, 85, 2]$ il risultato di questa funzione sarà $[2, 42, 54, 78, 1, 23, 85]$.

Esercizio 4

Implementare il Quicksort con *three-way partition*. L'algoritmo si differenzia dal Quick Sort per la fase di partizionamento. In questo caso la funzione `distribuzione` divide l'array in tre intervalli (invece di due): 1) gli elementi minori del pivot; 2) gli elementi uguali al pivot; 3) gli elementi maggiori del pivot. Ovviamente il valore restituito dalla funzione `distribuzione` non è più solamente la posizione del pivot dopo il partizionamento, ma anche il numero di elementi uguali al pivot. A partire dall'implementazione

dell'esercizio 1, scrivere un Quick Sort su interi che effettui la *three-way partition* su un array letto da input e ne stampi il risultato.