**Esercizio 1: Merge Sort e Quick Sort**

Note di implementazione

L’input del campo da ordinare avviene da tastiera a tempo di esecuzione, mentre la selezione dell’algoritmo è effettuata automaticamente (mergesort per int e float, quicksort per stringhe). è possibile selezionare l’algoritmo manualmente cambiando la macro AUTO da 1 a 0  
L’implementazione di quicksort era inizialmente più efficiente (nell’ordine dei 25s per int e float), ma non sono riuscito per qualche motivo a far funzionare il sort sulle stringhe sul totale dei record (su input minori funzionava). Quindi sono passato a un’implementazione di quicksort tripartita, che seppur meno efficiente funziona in tutti e tre i casi

L’esecuzione è cronometrata automaticamente tramite un timer interno realizzato con la libreria time.h

Tempi di esecuzione (+/- 2s):

|  |  |  |  |
| --- | --- | --- | --- |
| **Algoritmo/Tipo** | Int | String | Float |
| Merge Sort | 30s | 44s | 34s |
| Quick Sort | 43s | 47s | 45s |

Assieme all’algoritmo sono stati aggiunti i gli unit test, il cui scopo e’ quello di testare il corretto funzionamento di tutti i casi limite (test su elementi vuoti, un elemento, elementi uguali o due elementi diversi) per verificare la la correttezza degli algoritmi implementati. I test utilizzano unity, i cui file sono stati inseriti nella cartella del codice.