

# Fondamenti di Informatica II

---

## Esercitazione del 25 Febbraio 2020

---

**Attenzione:** L'esercitazione può essere svolta sia in linguaggio C che in Java. Il codice delle classi di base da cui iniziare è per tale ragione fornito sia nella versione Java che in C. Il codice è accessibile dalla home page del corso, insieme al testo dell'esercitazione.

**Obiettivi.** Gli obiettivi di questa esercitazione sono i seguente: i) prendere confidenza con l'ambiente virtuale BIAR; ii) (ri)visitare semplici comandi della shell Linux/Unix; iii) modificare, compilare ed eseguire codice usando il terminale a riga di comando; iv) usare makefile per gestire il processo di compilazione (C); v) passaggio di argomenti a riga di comando (Java e C).

### Task 1. Eseguire codice Java o C a riga di comando

Prendere confidenza con l'ambiente (macchina virtuale BIAR). Familiarizzare con i comandi elementari della shell linux: `ls`, `cd`, `find`, `grep`, `cp`, `mv`, `rm`, `mkdir`, `rmdir`, `man`, `cat`, `chmod`. Abituarsi a lavorare sul terminale: editing, compilazione ed esecuzione. Utili strumenti: redirectione input/output e file/command completion.

- Per l'editing si consiglia di usare l'editor di testo **Geany**. Chi preferisce può usare **vim**. Si raccomanda di non usare IDE in questa fase.
- Per la compilazione: usare `gcc / javac`. In Java: `javac *.java`; in C: `gcc *.c` (provare successivamente a usare il `makefile`)
- Per l'esecuzione: invocazione a riga di comando (C; assicurarsi che il risultato della compilazione sia eseguibile; se non, usare il comando `chmod` per renderlo tale) oppure `java <class file>`

Usare l'esempio di Bubble Sort (il codice è fornito). Nella versione di base, l'array da ordinare è definito staticamente all'interno del codice.

### Task 2. Estensioni

Estendere il codice in modo da leggere l'array da ordinare direttamente da riga di comando (`argc/argv` in C o `args[]` in Java) oppure generando un array di interi casuali di dimensione data. Nel secondo caso gli argomenti passati al programma saranno la stringa "rnd" seguita da spazio e da un intero che specifica la dimensione dell'array. In C la generazione di un array casuale di dimensione data è già realizzata dalla funzione `randArray`, in Java occorre usare la classe `java.util.Random`, dopo averne controllato la documentazione. Si aggiunga poi del codice che consenta di misurare il tempo di esecuzione del BubbleSort. Tale funzionalità può essere implementata in C attraverso la funzione di libreria `clock()` (cfr. ad esempio [https://www.gnu.org/software/libc/manual/html\\_node/CPU-Time.html](https://www.gnu.org/software/libc/manual/html_node/CPU-Time.html)), mentre in Java si può implementare usando ad esempio `System.currentTimeMillis()` (verificare la documentazione).