

Appunti di Laboratorio di Calcolo

Liam Ferretti

21 ottobre 2025

Sommario

Affinché si possa svolgere la prova di esonero bisogna aver non più di 3 assenze in laboratorio. È necessario installare sul proprio computer un compilatore C gcc, e un interprete Python 3.x. Per i ricevimenti con Boeri si possono effettuare il mercoledì dalle 11 alle 12 nella stanza 407 al quarto piano dell'edificio "Fermi", per Spagnolo il martedì dalle 12 alle 13 nella stanza 116 al primo piano dell'edificio "Fermi". Si raccomanda l'acquisto del libro di testo: "Programmare la scienza L.Barone", che si potrà portare all'esame. Il laboratorio Pontecorvo si trova in via Tiburtina 205.

Esame sarà una prova pratica di 3 ore, ma si potranno avere punti bonus ottenuti per chi frequenta il laboratorio all'esonero, il punteggio bonus è definito dal voto dell'esonero:

- 18-23: 1 punti
- 24-25: 2 punti
- 26-27: 3 punti
- 28-29: 4 punti
- 30/30 e lode: si è esonerati dalla prova finale.

Gli argomenti del corso saranno:

- Parte 1: Come funziona un computer
- Parte 2: Elementi di base del C
- Parte 3: Elementi avanzati del C
- Parte 4: Applicazioni e algoritmi + Python

Indice

1 Teoria del C	3
1.1 Scrivere un programma in linguaggio in C	3
1.1.1 Dichiarazione delle variabili	4
1.2 Assegnazione delle variabili	4
2 Operatori	5
3 Stringhe di formato	5
4 Istruzioni UNIX	6

1 Teoria del C

Si dice

- Procedurale, ovvero, è un linguaggio che lavora con una sequenza di istruzione che vengono eseguite in un ordine ben preciso, ed è possibile dividerle in sotto blocchi di codice, procedure/ funzioni, che svolgono una specifica parte del calcolo.
- Non completamente strutturato, usa alcune regole di base, che si basano su altre regole, e nel C si posso "superare" queste regole.
- Tipizzato, si devono dichiarare le variabili, e bisogna specificare il tipo di variabile, ovvero che tipo di valore può assumere la variabile.

I sistemi operativi sono organizzate in cartelle, ovvero directory, divise in una struttura ad albero, dove all'apice si trova la cartella base, ovvero in linux "/", in windows si trova con il nome "C:". Di base in linux è divisa in sottocartelle in cui si trovano i dati principali e più importanti del computer, una tra queste è la home, dove si trovano le sottocartelle degli utenti. Ogni file ha un certo percorso (path), ad esempio:

/home/utente/nomefile.estensione

È importante specificare l'estensione del file, e identifica il tipo di file.

1.1 Scrivere un programma in linguaggio in C

Serve un editor di testo, ovvero un programma che permette di scrivere testo in un file, ed è diverso da un word-processor, come Word, ad esempio EMACS, che produce file testuali con estensioni diverse, che contiene solo i caratteri, senza le informazioni sul tipo di carattere, grandezza, posizione etc etc.

Per scrivere un programma si parte dal file, in formato ".c", una volta scritto il codice, si usa il compilatore per generare il file con formato ".o", poi avviene la fase di linking, che "linka" le librerie, generando il file .x, la fase di compilatore e linker avviene insieme. Prima della fase di compilazione avviene la pre-processione.

```
1 #HEADER (tra cui #include<math.h>)
2
3 int main(){
4     //contiene le istruzioni del programma principale, definito corpo
5     // principale del programma
6 }
```

L'HEADER (direttive per il pre-processore, prende le istruzioni), il pre-processore sostituisce l'header con il codice delle librerie.

Il corpo principale del programma inizia sempre con la dichiarazione delle variabili, seguite poi dalle istruzioni effettive del codice, ogni istruzione ha alla fine ";", per specificare che l'istruzione è finita.

Un programma è composto da:

- Header: inclusione delle direttive, come le librerie ad esempio include<stdio.h>, o il DEFINE per definire le costanti simboliche, in questo modo ogni volta che nel programma si incontra la costante, verrà sostituita con il suo valore.

Nell'header si dichiarano anche le funzioni che verranno poi usate nel file.

- Body: ovvero la parte del programma compresa tra le graffe dell'int main
- Funzioni

Si possono compilare più file, concatenandoli in successione prima di "-o" all'interno del comando per compilare, e produce un singolo eseguibile.

1.1.1 Dichiarazione delle variabili

```

1 int main(){
2
3     int a= 0,b,c;
4     float f1, f2, f3;
5     double d1, d2, d3, d4;
6 }
```

Bisogna specificare il tipo ed il nome della variabile, esistono diverse tipologie di variabili:

- **int**: intero (16bit)
- **float, double**: numero razionale, in virgola mobile (32bit, 64bit)
- **char**, singolo carattere (8bit)
- Classificatori:
 - short
 - long

1.2 Assegnazione delle variabili

```

1 int main(){
2
3     int a = 0,b = 0,c = 0;
4     float f1 = 0, f2 = 0, f3 = 0;
5     double d1 = 0, d2 = 0, d3 = 0, d4 = 0;
6
7     //il segno = è l'operatore di assegnazione, assegna alla variabile a
     //sinistra l'elemento a destra, è comune assegnare 0 alla
     //variabile in quanto gli spazi di memoria potrebbero essere già
     //stati usati
8
9     return 0;
10 }
```

2 Operatori

Operazione	Simbolo
Assegnazione	=
Somma	+
Sottrazione	-
Moltiplicazione	*
Divisione	/
Modulo	%
Incremento	++
Decremento	--
Assegnazione con somma	+=
Assegnazione con sottrazione	-=
Confronto	==
Maggiore uguale	>=
Minore uguale	<=

Tabella 1: Operatori aritmetici e di assegnazione

3 Stringhe di formato

- %d, %i, %x — Interi (int) Esempio: `printf("Valore: %d", 10);`
- %f, %lf, %g, %e — Numeri reali (float / double) Esempio: `printf("Pi: %f", 3.14);`
- %c — Caratteri (char) Esempio: `printf("Lettera: %c", 'A');`
- %s — Stringhe (string) Esempio: `printf("Stringa: %s", "Ciao");`
- n: nuova linea
- t: tab
- v: vertical tab
- b: backspace

4 Istruzioni UNIX

Elenco comandi utili in UNIX

- Per creare una cartella: `mkdir nomecartella`
- Per entrare in una cartella: `cd directorydellacartella`
- Per tornare uscire da una cartella: `cd ...`, per tornare alla home directory: `cd`
- Per vedere il contenuto di una directory: `ls`