

## MODULO DI: FONDAMENTI DI INFORMATICA C

### Docente:

Franco Zambonelli

### Esercitatore:

Costantino Grana

### Come Contattarci:

#### E-mail (consigliato)

franco.zambonelli@unimo.it

grana@dsi.unimo.it

#### Telefono

059-2056133

### Ricevimento

Martedì pomeriggio dopo la lezione

Dipartimento di Scienze dell'Ingegneria

Via Vignolese 905, primo piano

### Materiale sul Corso (lucidi e informazioni):

<http://www.dsi.unimo.it/Zambonelli>

### Libro Consigliato:

D. Beneventano, S. Bergamaschi, C. Sartori,  
"Fondamenti di Informatica: Linguaggio di  
Programmazione C", Progetto Leonardo, Ed.  
Esculapio, 2000.

Introduzione

## PROGRAMMA FOND. INF. C

### Gestione Applicazioni e Strutture Dati Complesse:

- Dati Strutturati in C: Record e Puntatori
- I File in C
- Progetto multi-modulo dei programmi
- Studio della Complessità
- Gestione di Liste Dinamiche e Alberi in C

(gli argomenti includono i capitoli 9-10-11-12-13 del libro di testo consigliato)

- Dalla programmazione modulare alla programmazione ad oggetti

## REGOLE ESAME

### Quiz + Prova al Calcolatore + Discussione

Consiglio: studiare POCO su libri e appunti e fare invece MOLTA pratica al calcolatore!!!!

Introduzione

## PERCHÉ QUESTO CORSO???

### Fondamenti A e B:

- acquisizione nozioni base di informatica
- strumenti per la definizione di semplici algoritmi per la elaborazione di semplici strutture dati

### Fondamenti C e D:

- simpaticamente, il mondo reale presenta strutture dati complesse e algoritmi complessi, in un ambiente in continua evoluzione.
- Il calcolatore, deve supportare l'uomo nella gestione delle problematiche del mondo reale!
  - Strutture dati complesse e dinamiche
  - Algoritmi complessi!

### Dovremo quindi:

- acquisire nozioni e strumenti più avanzati di programmazione
- esercitarci nella gestione di strutture dati complesse
- riuscire a valutare la complessità computazionale dei programmi
- imparare metodologie per la riduzione della complessità progettuale dei programmi!!!!

Introduzione

## ESEMPI

### Descrizione Struttura Organizzativa Aziendale

Che struttura dati ci serve per rappresentarla?

- Non certo vettori o stringhe....é una struttura dati
- complessa, un grafo di relazioni e responsabilità....
- Inoltre, è dinamica, perché i membri della organizzazione vanno vengono e cambiano ruolo, e cambiano in numero....

Che algoritmi bisogna definire per gestire tali strutture e i loro cambiamenti?

### Sistema di gestione a "numeri delle code" in banca:

- Che struttura dati la può rappresentare?
- Come gestiamo l'arrivo di nuovi clienti e il fatto che un cliente venga servito, uscendo dalla fila?
- Come gestiamo il fatto che ci potrebbe essere una gestione dei clienti privilegiati nella lista?

### Banche dati DNA:

- il DNA ha struttura elicoidale complessa
- come memorizziamo i dati del DNA?
- che algoritmi ci servono per elaborare i dati memorizzati ed estrarne informazioni?

Introduzione

## ALLARGANDO LA VISIONE....

Informatica è la scienza della “elaborazione automatica dell’informazione”

Il concetto di “elaborazione della informazione” è pervasivo nell’universo, e anche nelle nostre attività cognitive e attuative.

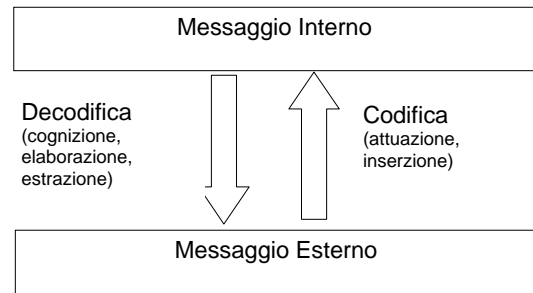
Il concetto stesso di informazione perde di senso senza gli algoritmi che la “elaborano”.

Esempi:

- un pentagramma è informazione musicale? No, senza i processi cognitivi e attuativi che dalla partitura portano alla produzione di musica reale.
- il DNA è informazione su un essere vivente? NO, senza i processi bio-chimici che dal DNA formano un individuo.
- Un tuono è informazione su un temporale in arrivo? NO, senza i processi cognitivi che ci portano a dire “non è Giove che si è arrabbiato ma è solo il rumore di una scarica elettrica dovuta a una differenza di potenziale nell’atmosfera tipicamente associata a precipitazioni”

Introduzione

## IL CONCETTO DI INFORMAZIONE



L’informazione “utile” si produce realmente a livello di messaggio esterno.

Esempio:

DNA → messaggio interno  
Processi chimico biologici → decodifica  
Individuo → messaggio esterno

Riusciamo a capire come funziona? No, i messaggi interni e la decodifica sono troppo complicati!

Esempio:

$3x - 6 = 0$  → messaggio interno  
regole algebriche → decodifica  
 $x = 2$  → messaggio esterno

Introduzione

## IL MESSAGGIO INTERNO

L’informazione spesso è condivisibile, trasmissibile, o disponibile, solo a livello di messaggio interno.

Esempi:

- Trasmissione caratteri ereditari → DNA
- Conoscenze matematiche → formule
- Memorizzazione stabile di musica nel 1800 → partitura

Quindi:

Necessità di trasposizione di informazioni in una forma “forma interna” che ne faciliti la

- trasmissione
- elaborazione
- ristrutturazione

In ogni caso, motivi di economia intellettuale e fisica suggeriscono di:

- minimizzare per quanto possibile la distanza tra la rappresentazione esterna e la rappresentazione interna e gli altri sforzi collegati!

Esempio:

partitura musicale: astrazione delle corde del liuto

Introduzione

## IL RUOLO DEI CALCOLATORI

Trasposizione della informazione relativa a oggetti/entità/strutture del mondo reale in una forma interna:

- a basso livello: forma digitale
- ad alto livello: forma tipica del linguaggio di programmazione utilizzato

Calcolatore come entità per:

- elaborare dalla forma interna a quella esterna
  - memorizzare
  - trasmettere
  - comprimere
  - rendere disponibile in forma esterna
- Informazioni del “mondo reale”

O, più’ in generale:

per velocizzare ed estendere i processi cognitivi di elaborazione (codifica e decodifica) e memorizzazione dell’informazione che noi siamo comunque in grado di compiere!

Introduzione

## IL RUOLO DEI PROGRAMMATORE

Nel suo lavoro di:

- decidere come codificare informazione in forma interna
- definire “algoritmi” per la trasposizione dalla forma interna a quella

Deve avere come obiettivi:

- minimizzare la distanza tra la forma interna e quella esterna (minimizzazione dello sforzo concettuale)
- minimizzare la complessità computazionale (sforzo algoritmico)
- minimizzare la complessità progettuale (sforzo di produzione del sistema software)

*Il buon programmatore lavora poco!*

In verità, il lavoro è semplice, perché non rappresenta altro che una trasposizione dei nostri processi cognitivi e attuativi.....(per molti aspetti, noi siamo degli elaboratori!).

Scopo del corso, sotto questa luce:

- acquisire le conoscenze “tecnologiche” necessarie per codificare ed elaborare le informazioni e le strutture del mondo, comunque complesse, in una forma interna elaborabile dal calcolatore
- studiare la codifica e la elaborazione di alcune strutture “pervasive”, quali le liste e le relazioni gerarchiche!