

E-Learning:
un esperimento via web su corsi di
Fondamenti di Informatica

Progettare in modalità e-learning con focus sul discente

Domenico Capano

Dicembre 2004

*A Maria Giovanna Capomolla e
Gregorio Capano, miei genitori*

Indice

Alcune considerazioni sul testo	9
INTRODUZIONE.....	11
1 Come si costruisce un generico Corso ODL.....	13
1.1 Che cosa è l'e-Learning	13
1.2 I Learning Objects	16
1.2.1 Caratteristiche dei Learning Objects	16
1.2.2 I metadata	18
1.3 Generalità sulla progettazione di un Corso ODL	22
1.3.1 Una metodologia di progetto di un generico corso	25
1.3.2 Metodologia didattica di assegnazione delle prove intermedie	32
1.3.3 Metodologia di assegnazione dei crediti formativi nelle prove intermedie.....	34
1.3.4 Applicazione delle formule al corso FIODL.....	39
1.3.5 Obiettivi formativi.....	47
2 L'esperimento eseguito: Corso FIODL.....	49
2.1 Ambiente di lavoro	49
2.2 Obiettivi ed aspetti dell'esperimento	50
2.3 Organizzazione	52
2.3.1 Lista di iscrizione e conoscenza iniziale	53
2.3.2 Elenco degli iscritti ai corsi.....	55
2.3.3 Test Introduttivi.....	56
3 Svolgimento del Corso	60
3.1 Lezioni.....	60
3.2 Comunicazione.....	61
3.2.1 Stato del Corso	65
3.2.2 Autovalutazione dei partecipanti al corso	66
3.2.3 Valutazione degli studenti.....	66
3.2.4 Valutazione del corso	67

3.3	Test di autovalutazione.....	67
3.4	Prove intermedie assegnate	71
•	Soluzione della Prova Intermedia 7 per F1 proposta nel corso1,1 .74	
3.4.1	Cartella personale dello studente S3,1 protetta da password	75
3.4.2	Status di valutazione personale	76
3.5	Test Random.....	76
3.6	Area Riservata	79
3.7	Test di valutazione finale.....	82
4	Risultati prodotti dal corso.....	85
4.1	Grafici di analisi dei risultati forniti dall'esperimento	85
4.2	Generazione crediti formativi in FIODL	89
4.3	Risultati del Questionario di valutazione del corso FIODL	95
5	Come puntare il focus sull'apprendimento.....	99
5.1	Path learning.....	99
5.2	Tipologie di livelli di apprendimento e path learning statico	101
5.3	Diagramma di flusso arricchito con caratteristiche adattative “costruzione corso”	103
5.4	Esempio chiarificatore delle formule 1.20 e 1.21	109
5.5	Calcolo numero stati del ramo di Riassemblaggio RSMOD	109
5.6	Diagramma ASM per automatizzare il corso	112
6	Ruolo delle ontologie nei corsi e-Learning.....	117
6.1	Introduzione.....	117
6.2	Ontologie.....	119
6.3	Cenni sul semantic web	121
7	Appendice A -Test finale corso	124
7.1	Test finale del corso di FI-ODL	124
8	Appendice B.....	136
8.1	Questionario di valutazione del Corso FIODL	136
8.1.1	Risultati Valutazione del Corso FI-ODL	141
8.2	Un esempio di risposta al questionario di valutazione di FIODL pervenuta.	146
9	Appendice C - Lezione XVI e commenti sulla sua organizzazione	148
9.1	Ordinamento e ricerca	148
9.2	Ordinamento Con Selection Sort.....	149
9.2.1	Algoritmo ordinamento per selezione	149
9.3	Ordinamento a bolle	162
9.4	Ordinamento di altre strutture dati	165
9.5	Ricerca di un elemento in un array di interi	167
9.5.1	Ricerca sequenziale.....	167
9.5.2	Ricerca binaria in un array di numeri interi	168
9.6	Videata del link del sito Test 16	174

9.6.1	Test di autovalutazione 16, dopo aver risposto a tutte le domande esattamente.....	175
9.6.2	Test di autovalutazione 16 con tempo impiegato per la sua risoluzione.....	176
10	Appendice D.....	177
10.1	I Learning Objects della Lezione 16	177
10.2	I Learning Objects extra della L16.....	186
11	Appendice E.....	188
11.1	Contenuto delle lezioni ed attività didattiche correlate	188
11.1.1	Soluzione proposta alla Prova Intermedia 3 di F2	197
11.1.2	Codice per effettuare l'invio a tutta la mailing list.....	199
•	Bibliografia	202
•	Sitografia.....	202

Indice delle Figure

Fig. 1.1 - (Elementi dello standard Dublin Core)	19
Fig. 1.2 - (Percorso di conoscenza).....	23
Fig. 1.3 - (Fasi di soluzione di un problema).....	23
Fig. 1.4 - (Livelli di una Biblioteca)	25
Fig. 1.5 - (Corrispondenza fra L.O. di tipo test e L.O. di tipo Lezione)	43
Fig. 1.6 - (Prova Intermedia 3 per F2)	44
Fig. 1.7 - (Diagramma di flusso seguito dal learner nel corso F2).....	44
Fig. 2.1 - (Home page corso FIODL)	49
Fig. 2.2 - (Home page sito web del Consorzio NETTUNO)	51
Fig. 2.3 - (Frameset FIODL).....	52
Figure 2.4 - (Parte del modulo di iscrizione 2)	54
Fig. 2.5 - (Test Introduttivo F1 (a)).....	57
Figure 2.6 - (Test Introduttivo F1 (b, c))	57
Figure 3.1 - (Lezioni e date della loro pubblicazione sul web).....	61
Fig. 3.2 - (Invio E-mail ad utente selezionato dal menu a tendina)	62
Fig. 3.3 - (Invio E-Mail ad alcuni utenti scelti)	62
Fig. 3.4 - (Invio E-Mail a tutta la Mailing list)	63
Fig. 3.5 - (Chat non vocale)	64
Fig. 3.6 - (Modello di comunicazione)	64
Fig. 3.7 - (Stato del corso)	65
Fig. 3.8 - (Videata Questionario di valutazione Corso FIODL)	67
Fig. 3.9 - (Test a completamento).....	68
Fig. 3.10 - (Test ad opzioni)	69
Fig. 3.11 - (Test a risposta multipla).....	70
Figure 3.12 - (Elenco Prove del corso FIODL)	71
Fig. 3.13 - (Cartella personale del learner S3,1)	75
Fig. 3.14 - (Videata dei links ai test random F2)	76
Fig. 3.15 - (Domande Random di ricapitolazione F2 (A)).....	77

Fig. 3.16 - (Domande Random di ricapitolazione F2 (B)).....	77
Fig. 3.17 - (Domande Random Test 17 (A)).....	78
Fig. 3.18 - (Domande random Test 17 (B))	79
Fig. 3.19 - (Area riservata utenti FIODL).....	80
Fig. 3.20 - (Esempio di contenuto della cartella personale del docente)	81
Figure 3.21 - (Prova Finale Fondamenti 2).....	83
Fig. 4.1 - (Voti Medi Prove F1 (media semplice)).....	85
Fig. 4.2 - (Voti Medi Prove F2 (media semplice)).....	86
Fig. 4.3 - (Voti Test Introduttivo e Prova Finale)	87
Fig. 4.4 - (Voti Test Finale, Introd. e Prove intermedie)	88
Fig. 4.5 - (Crediti formativi, Voti Test Finale e Voto Esame)	94
Fig. 4.6 - (Modello ADDIE)	97
Fig. 4.7 - (Modified ADDIE Model)	98
Fig. 5.1 - (Path Learning).....	100
Fig. 5.2 - (Schema di selezione tipologia corso).....	102
Fig. 5.3 - (Diagramma di flusso contenente il focus sull'apprendimento)....	107
Fig. 5.4 - (Caselle ASM).....	112
Fig. 5.5 - (Schema di un Multiplexer).....	113
Fig. 5.6 - (Diagramma ASM parziale di Mealy adattativo)	115
Fig. 9.1 - (Videata Lezione XVI).....	149
Fig. 9.2 C - (Traccia esercizio selection sort)	154
Fig. 9.3 C - (suggerimento esercizio selection sort)	155
Fig. 9.4 C - (soluzione proposta esercizio selection sort)	156
Fig. 9.5 C - (esempio di ordinamento interattivo con selection sort).....	157
Fig. 9.6 C - (Finestra interattiva Bubble Sort)	166
Fig. 9.7 - (Traccia Esercizio ricerca binaria)	172
Fig. 9.8 - (Soluzione esercizio ricerca binaria)	173
Fig. 9.9 C - (Test di autovalutazione 16)	174
Fig. 9.10 C - (Test di autovalutazione 16 risolto)	175
Fig. 9.11 C - (Test di autovalutazione 16 tempo impiegato)	176

Indice delle Tabelle

Tab. 1.1 - Risoluzione problema e progettazione corso.....	24
Tab. 1.2 - Generica Biblioteca divisa in Moduli.....	31
Tab. 1.3 - Esempio di Voti e Pesi Prove Intermedie.....	37
Tab. 1.4 - Moduli e SottoModuli F1	39
Tab. 1.5 - Moduli e SottoModuli F2	41
Tab. 1.6 - Divisione Modulo di F2 in SottoModuli e L.O.	43
Tab. 2.1 - Elenco iscritti corso FIODL	55
Tab. 4.1 - Moduli e SottoModuli di tipo Prove Intermedie F1	90
Tab. 4.2 - Voti Prove Intermedie del Learner S4,1	91
Tab. 4.3 - Moduli e SottoModuli di tipo Prove Intermedie F2	92
Tab. 4.4 - Voti Prove Intermedie del Learner S6,2.....	92
Tab. 4.5 - Valutazione corso learners F1 e F2	95
Tab. 5.1 - Calcolo fattore di normalizzazione.....	109
Tab. 5.2 - (Configurazione da esito del test di autovalutazione)	114
Tab. 10.1 - I learning Objects del Modulo 4 di F2.....	177
Tab. 11.1 - Numero di domande in FIODL e tipo test autovalutazione.....	196

Alcune considerazioni sul testo

Nel testo dopo aver dato una definizione, condivisa dalla comunità scientifica italiana, di e-Learning ed aver introdotto i concetti di Learning Objects e di Metadata, descriviamo una “metodologia” di progetto di un generico Corso online (*Titolo*), appartenente ad un certo Catalogo (*Corso di Laurea*) di una certa Biblioteca, che assimiliamo ad una (*Facoltà Universitaria*), facendo uso di Moduli (*Unione di Lezioni, Test di autovalutazione e Prove Intermedie* aventi un omogeneo fine didattico), SottoModuli (*Lezioni, Test autovalutazione e Prove Intermedie*) e Learning Objects.

L’obiettivo dell’esperimento via web condotto nella tesi di laurea in Ingegneria da cui questo libro è stato elaborato era duplice: stabilire un *modello d’erogazione di un corso online* ed analizzare, con dati statistici suffragati da grafici, *l’impatto pedagogico che la didattica a distanza aveva sui discenti*, traendo la tesi spunto dal progetto EUROCOMPETENCE (European Commission Socrates Programme).

Nel testo abbiamo realizzato una sistemazione metodologica dell’esperimento per precisare meglio le azioni condotte durante e (prima) esso.

Al fine di stabilire un “modello di descrizione/costruzione” per un corso ODL¹ web based abbiamo introdotto le seguenti definizioni:

Una Biblioteca (che assimiliamo ad una facoltà universitaria) è *un insieme di Cataloghi*.

Un Catalogo (che assimiliamo ad un corso di laurea della facoltà universitaria) è *un insieme di Corsi*.

Un Corso (che assimiliamo come un generico titolo di un corso di laurea) è *un insieme di Moduli*.

Un Modulo è un insieme di SottoModuli, trattanti una omogenea tematica didattica.

Un SottoModulo (che può essere o una lezione del corso o un test di autovalutazione od una Prova Intermedia) è *un insieme di Learning Objects*.

Precisiamo sin da subito che, nella pratica della costruzione dei corsi online, dobbiamo considerare un particolare Learning Object, il *L.O. domanda/risposte*. Si tratta di un learning object presente nei test di autovalutazione, che assumiamo essere la generica domanda e tutte le sue possibili opzioni risposta.

¹ Open and Distance Learning

Definiamo un test di autovalutazione come un SottoModulo contenente l'insieme di tutti i learning objects di tipo domanda/risposte in esso presenti.

Definiamo una Prova intermedia come un SottoModulo erogato alla fine dei SottoModuli compresi nel Modulo che la contiene.

Le caratteristiche di originalità, dei contenuti del testo, sono costituite dal fatto che le formule matematiche introdotte nel testo sono generali ed automatizzabili a qualsiasi corso online ed inoltre dal fatto che i SottoModuli di Tipo Lezione sono costruiti partendo dall'unione dei Learning Objects dei Test di autovalutazione corrispondenti didatticamente.

Inoltre nel Quinto capitolo, *Come Puntare il focus sull'apprendimento*, abbiamo introdotto un metodo per automatizzare un corso online sulle effettive esigenze del discente, basato sul metodo della correzione d'errore introdotto dalle analisi pedagogiche eseguite (sulle risposte date dai learners alla valutazione del corso e-Learning medesimo e sulle domande fatte o risposte date utilizzando le altre forme d'interazione presenti nel corso) ed abbiamo utilizzato un diagramma a stati ASM (Algorithmic State Machine) di Mealy adattativo per automatizzare la personalizzazione dell'apprendimento. Nel Sesto capitolo tratteremo brevemente del concetto di ontologia utilizzabile per creare corsi on-line in modalità e-learning e per il semantic web

II Edizione. Il libro, "E-Learning: un esperimento via web su corsi di Fondamenti di Informatica. Progettare l'elearning con focus sul discente", è disponibile soltanto per uso personale; tutti i diritti sono riservati e di proprietà dell'autore Domenico Capano.

Per eventuali chiarimenti sui contenuti del volume l'autore è disponibile e reperibile tramite l'email: mimmocapanoNOSPAM@comunedasa.it

Buona lettura!

INTRODUZIONE

Il testo, rielaborazione della tesi di laurea “*Open and Distance Learning via web: un esperimento su corsi di Fondamenti di Informatica*”, tratta di un esperimento di didattica a distanza “Fondamenti di Informatica in Open and Distance Learning” da noi condotto attraverso il web nel 2001 su un gruppo di dodici studenti iscritti al primo anno di Ingegneria (Consorzio NETTUNO).

Rispetto al 2001 la didattica a distanza, inizialmente osteggiata in tanti ambienti accademici e non, ha fatto notevoli progressi. Ad essa e più in generale all’e-Learning si avvicina, ogni giorno, un numero sempre maggiore d’esperti e formatori pubblici e privati. Internet e il Web forniscono, alle istituzioni che si occupano di formazione, un mezzo efficiente per distribuire didattica ed offrono grandi potenzialità d’interattività e di collaborazione. Oggi si parla tanto di *e-Learning* ed adesso siamo in pieno boom di esperimenti e pratica di corsi a distanza, realizzati nel pubblico come nel privato, aiutati in ciò sia dal progresso fatto nella tecnologia d’erogazione dei medesimi, sia dai danari elargiti dalle pubbliche istituzioni per questa nuova metodologia didattica, col fine ultimo probabilmente di vincere un’ipotetica *competitività globale* delle aziende italiane.

Si scommette quindi sulla *formazione continua* del personale, la cosiddetta *lifelong learning*², realizzabile in modo efficiente ed economicamente conveniente, per alcune scale numeriche e geografiche d’utenza, attraverso l’e-Learning.

L’esperimento da noi fatto nel web quindi è retrodatato rispetto alle attuali tendenze, (avendo però sin da allora presente che la principale caratteristica dei corsi online è la mancanza di rigidità fissate dal tempo e dagli spazi fisici all’interno dei quali avviene l’istruzione al fine di poter favorire studenti-lavoratori ed altri discenti che non possono frequentare i corsi tradizionali e, che la didattica a distanza pone al centro del modello d’apprendimento lo studente³) ma la parte compilativa di tal esperimento, oggetto di questo libro, è stata rivista alla luce di ciò che oggi si muove nel settore dell’e-Learning.

Nel primo capitolo dopo aver dato una definizione condivisa dalla comunità scientifica di e-Learning ed aver introdotto i concetti di *Learning Objects* (unità didattiche atomiche) e di *metadata*, descriviamo una “metodologia di progetto” di un generico corso online, appartenente ad un

² Formazione continua (Traslitterazione britannica di quell’*educazione permanente* che già aveva caratterizzato il pensiero pedagogico di *Johannes Amos Comenius* — *Didactica Magna*, 1635) di *Fabrizio Pecori*, <http://integrazione.mymedia.it/e-learning.pdf>

³ Teoria psicopedagogia del *costruttivismo*

certo catalogo di una certa biblioteca, che assimiliamo ad una facoltà universitaria, facendo uso di *Moduli*, *SottoModuli* e *Learning Objects*. Questa teoria è stata costruita sfruttando l'esperimento condotto nel web. Nel capitolo, inoltre, sono introdotte diverse formule matematiche che descrivono i vari livelli della nostra biblioteca e che applichiamo ai corsi erogati a distanza attraverso il web.

La teoria proposta la adeguiamo, al nostro esperimento FIODL⁴ nel capitolo suddetto e nei successivi, per intero.

Nel secondo capitolo illustriamo l'esperimento eseguito: ambienti di lavoro, obiettivi ed organizzazione.

Nel terzo capitolo mostriamo lo svolgimento del corso: le *lezioni*, la comunicazione, i tipi di *test di autovalutazione* usati nell'esperimento e le *prove intermedie*.

Nel quarto capitolo con l'ausilio di alcuni grafici illustreremo i risultati forniti dall'esperimento.

Nel quinto capitolo del libro si mostrerà una nostra "metodologia" di automatizzazione di un corso online, per puntare il *focus sull'apprendimento* (didattica, che pone al centro l'allievo, *learner centered* e non il docente come avviene nei corsi tradizionali), attraverso un diagramma a stati ASM.

Nel sesto capitolo introdurremo il concetto di ontologia e si mostrerà come è possibile, attraverso esse, l'introduzione di semantica nei learning objects per esplicitarne il contenuto con conseguente facilità di loro riusabilità in diversi contesti didattici; tratteremo brevemente inoltre del web semantico.

Le Appendici A, B, C, D, ed E sono poi incluse allo scopo di mostrare saggi del materiale didattico concretamente prodotto e degli aspetti metodologici coinvolti (applicati o solo progettati).

A e B contengono rispettivamente: il test finale utilizzato nel corso a distanza ed il questionario di valutazione del medesimo con un esempio di risposta pervenuta attraverso e-mail.

Nell'appendice C mostriamo un esempio di lezione a distanza ed i commenti sulla sua organizzazione.

Nell'appendice D mostriamo come tale lezione sia un aggregato di Learning Objects.

Nell'appendice E si mostrerà la quantità di lavoro svolto nel Corso FIODL.

⁴ Fondamenti di Informatica in Open and Distance Learning

1 Come si costruisce un generico Corso ODL

1.1 Che cosa è l'e-Learning

Nell'analizzare il settore dell'e-Learning è necessario innanzi tutto definire cosa s'intende oggi per e-Learning. Una definizione alquanto condivisa nella comunità scientifica italiana è quella proposta dall'osservatorio ANEE ⁵: «L'e-learning è una metodologia d'insegnamento e apprendimento che coinvolge sia il *prodotto* sia il *processo formativo*. Per prodotto formativo s'intende ogni tipologia di materiale o contenuto messo a disposizione in formato digitale attraverso supporti informatici o di rete. Per processo formativo s'intende invece la gestione dell'intero iter didattico che coinvolge gli aspetti di erogazione, fruizione, interazione, valutazione. In questa dimensione il vero valore aggiunto dell'e-learning emerge nei servizi di assistenza e tutorship, nelle modalità di interazione sincrona e asincrona, di condivisione e collaborazione a livello di community.

Peculiarità dell'e-learning è l'alta flessibilità garantita al discente dalla reperibilità sempre e ovunque dei contenuti formativi, che gli permettono

⁵ Associazione Nazionale dell'Editoria Elettronica. L'Osservatorio E-learning 2003 di ANEE, con il patrocinio del Ministro dell'Innovazione e delle Tecnologie, è stato realizzato con la collaborazione di alcune tra le principali aziende Italiane operanti nell'ambito della formazione stanziale e a distanza e di alcune tra le Università caratterizzate dalla maggiore propensione all'innovazione: Microsoft, Banca Intesa, Sfera, Telecom Italia Learning Services, Isvor Fiat, E-Ducation, Università Statale di Milano – CTU, Politecnico Milano, LUISS G. Carli, Fondazione Methis. Obiettivo dell'Osservatorio ANEE è fornire uno scenario aggiornato del contesto competitivo del settore in termini di caratteristiche dell'offerta, della domanda e delle tendenze in atto della formazione on line in Italia, con un approfondimento sul mondo dell'Università, sia come fornitore che utilizzatore di contenuti, e della Pubblica Amministrazione centrale e locale.

l'autogestione e l'autodeterminazione del proprio apprendimento; resta tuttavia di primaria importanza la scansione del processo formativo, secondo un'agenda che responsabilizzi formando e formatore al fine del raggiungimento degli obiettivi didattici prefissati».

Secondo l'Osservatorio ANEE si sta realizzando una vera e propria rivoluzione che cambierà il nostro modo di apprendere e di lavorare con inevitabili ripercussioni sulla struttura stessa dell'organizzazione del lavoro.

Dalle aziende alle istituzioni pubbliche, dalle scuole alle università, dalle iniziative dei comuni ai programmi europei, l'elearning è il fondamento su cui si costruirà il futuro della formazione e dell'aggiornamento professionale.

Nel convegno su "e-Learning Stato dell'arte e prospettive di sviluppo"⁶, svoltosi il 30 giugno 2003 all'Università degli Studi di Milano, Luciano Battezzati diceva: «Il mercato dell'e-Learning è un mercato ricchissimo ma molto faticoso. Oggi le aziende sempre di più ci richiedono di essere accompagnate nel processo di consulenza, ossia nel processo di ridisegnare il proprio processo formativo. [...] . Occorre quindi ripensare tutto il processo di gestione delle risorse umane. L'e-Learning di oggi deve farsi carico del processo di formazione continua che il mercato attuale richiede e ciò può farsi soltanto se si ripensa all'intero modo di produrre formazione (puntando sulla qualità). [...] . E' richiesta quindi una competenza dei formatori di alto livello, difficile da creare, difficile da trovare»⁷.

L'e-Learning è passato, quindi, dalla semplice gestione dei contenuti a soluzioni che integrano i sistemi di *knowledge management* o gestione delle conoscenze.

Il focus nell'e-Learning deve essere comunque sulle soluzioni di apprendimento non sulle tecnologie, «le tecnologie si dà per scontato che ci debbano essere»⁸. Riguardo a quest'ultima affermazione, la realtà strutturale italiana, in fatto di tecnologie (ad esempio larga banda), è a macchia di leopardo. Vi è l'intero mezzogiorno (in ritardo rispetto al centro-nord) che sta appena iniziando a dotarsi tecnologicamente, anche se ciò è visto dall'ANEE come una risorsa poiché vi saranno capitali investiti nel settore e-Learning in contemporanea al colmarsi del gap strutturale suddetto. Roberto Liscia, Presidente della Commissione ANEE, rilevava nel convegno citato, con dati

⁶ Convegno in formato web video al link: <http://www.ctu.unimi.it/osservatorioanee/evento.html>

⁷ Luciano Battezzati, ISVOR FIAT, e-LEARNING: CONVEGNO STATO DELL'ARTE E PROSPETTIVE DI SVILUPPO I RISULTATI DELL'OSSERVATORIO ANEE 2003 con il patrocinio del Ministro per l'Innovazione e le Tecnologie - Università degli Studi di Milano Lunedì, 30 giugno 2003, <http://www.ctu.unimi.it/osservatorioanee>

⁸ L. Battezzati, ISVOR FIAT, e-LEARNING: CONVEGNO cit.

statistici alla mano, che: «fra le aziende, da loro intervistate, dotate di banda larga attualmente (nel 2003) il 15% fa uso di e-Learning in modo sistemico e che nel 2004 dovrebbe passare al 46%». I dati ANEE resi noti nell'ultimo convegno del 27 settembre 2004 confermano questo trend di crescita.

Tuttavia pensiamo che se non si avvia, da subito, un piano di cablatura in fibra ottica, con intervento pubblico per l'Italia intera, le cosiddette *autostrade digitali* resteranno una chimera per i piccoli centri ancora per molti decenni e ciò, nel tempo in cui la società contemporanea sta passando progressivamente da società dell'informazione a *società della conoscenza*, non può che causare un ulteriore incremento della loro marginalizzazione economico-culturale.

Vista la probabile antieconomicità della cablatura in fibra ottica, connettere alcuni comuni interni o a bassa densità di popolazione e quartieri periferici delle città, significherebbe dover e poter utilizzare il satellite, come mezzo di comunicazione dati, per quei luoghi.

Per gli utenti della fascia consumer l'invio dati, dal navigatore al provider, avviene attraverso linea telefonica, se si ha il proprio pc collegato ad una particolare scheda ed all'antenna satellitare, la ricezione avviene attraverso satellite.

Le imprese e la P.A. possono utilizzare il satellite anche in trasmissione necessitando di larghezza banda superiore in generale.

1.2 I Learning Objects

I *learning objects* sono “blocchi” autonomi tra loro ed indipendenti dal contesto che possono essere assemblati (tra loro) in ogni momento in cui ciò è richiesto ed in base alle esigenze del discente⁹. In giro circolano tante definizioni dei Learning Objects, la più accreditata è quella data da David Wiley che definisce i Learning Objects come “*ogni risorsa digitale che può essere riutilizzata per supportare l'apprendimento*”¹⁰.

In questa definizione è inclusa qualsiasi cosa che può essere resa disponibile on-line; quindi i learning objects li possiamo pensare come "atomi di conoscenza" autoconsistenti.

Oggi giorno i soggetti erogatori di formazione stanno passando alla modalità di erogazione del materiale didattico basata sull'approccio “Learning Object” per via del più alto livello qualitativo riconosciuto ad esso rispetto ai metodi precedenti; inoltre i Learning Objects possono essere *riusati* se opportunamente specificati e realizzati¹¹.

1.2.1 Caratteristiche dei Learning Objects

Le principali caratteristiche che devono possedere i Learning Objects sono due la **combinazione** e la **granularità**; della combinazione si occupano i “computer agent” che possono comporre in modo automatico ed in modo dinamico le “lezioni”, personalizzandole per i singoli utenti. I Learning Objects grazie al fatto di avere i metadata che li esplicitano (come una specie di “dati di targa”), possono essere individuati, aggiornati e ricomposti, dai LCMS, per formare nuovi “moduli didattici” rispondenti in modo innovativo agli obiettivi formativi dei learners (percorso di apprendimento personalizzato e riadattabile dinamicamente alle esigenze del learner), nel rispetto di alcuni vincoli imposti dalla strategia di obiettivo formativo delle aziende, università ed enti cui il learner “dipende”.

⁹ Federica Bianchi, articolo tratto dalla tesi discussa da F. Bianchi nel 2002 presso l'Università degli Studi di Torino (relatore prof. Alessandro Perisinotto).

¹⁰ Wiley, David A., nel sito <http://wiley.ed.usu.edu/index.html>

¹¹ Per informazioni sui modi di etichettare i learning objects si veda il sito del LOM Working Group al link: LOMWG IEEE, <http://ltsc.ieee.org/wg12/>

La granularità è una caratteristica che esprime quali dimensioni (quanto deve essere grande) debba avere un Learning Object. Vi è abbastanza aleatorietà in ciò e, possiamo affermare che una risorsa per attribuire una dimensione al L.O. possa essere sicuramente il buon senso¹² (Argomenti diversi hanno difficoltà diverse) dell'*instructional designer* che è colui che immette i vincoli di selezione dei Learning Objects all'interno del software LCMS nel rispetto degli obiettivi didattico/formativi prefissati.

Maggiore è la combinabilità e la granularità dei L.O. definiti, migliore sarà la loro **riusabilità** (altra caratteristica alla base della scomposizione in blocchi logici) in diversi contesti didattici (**adattabilità**), più facile sarà la loro **localizzabilità** e più agevole sarà il loro uso personalizzato alle caratteristiche del learner, studente o discente.

¹² I learning Object non debbono superare la durata di 20 minuti coincidente con il valore medio della curva di attenzione del learner, un SottoModulo Lezione composto con combinazioni di Learning Objects deve avere durata inferiore alle due ore, vedi Dalla FaD all'e-Learning: la formazione a distanza non è più così distante, aut. Federica Garbolino, <http://www.isvor.it/aspasia.pdf>, pag. 20.

1.2.2 I metadata

L'approccio basato sui "Learning Objects" è sempre più diffuso. Standard internazionali come **AICC**¹³, **IMS**¹⁴ e **SCORM**¹⁵ stanno risolvendo il problema della *interoperabilità* fra le diverse piattaforme di formazione a distanza (**LMS**, Learning Management System).

Creare Learning Objects efficaci e realmente riutilizzabili, però, non è solo un problema tecnologico: richiede una specifica metodologia di progettazione e realizzazione, seguendo la quale, è possibile realizzare in tempi rapidi e a costi contenuti L.O. d'elevata qualità.

Per garantire che i learning objects siano aggregati, suddivisi e riutilizzati è necessario standardizzare la loro descrizione ovvero definire quello che in gergo si chiama "set di metadati".

I **meta-dati** (o metadata o Learning Object Metadata **LOM**) sono definiti come quei dati che non si riferiscono direttamente ai contenuti concreti di un L.O., ma li classificano, nel senso che rinviano ai contenuti di apprendimento del L.O.

Per esempio in una scheda di identificazione di un testo in una biblioteca sono meta-dati i campi: "autore", "titolo" etc.

Essi rinviano ai dati contenuti nei relativi campi per esempio "Settembrini" e "Ricordanze della mia vita" questi direttamente connessi ad una determinata opera letteraria.

¹³ Dal glossario di www.asfor.it, Acronimo di Aviation Industry CBT Committee, è un'associazione internazionale di professionisti di formazione basata sulle tecnologie CBT. Definisce linee guida e suggerimenti (AGR = AICC Guidelines and Recommendations) per ciò che concerne lo sviluppo e la distribuzione di prodotti e tecnologie per la formazione a distanza (vedi Fad) e certifica l'aderenza dei prodotti alle linee guida. Queste specifiche consentono una comunicazione tra i corsi e il sistema di gestione della formazione e permettono di rendere i corsi portabili da un sistema all'altro: infatti, tutti gli LMS che aderiscono allo standard AICC sono in grado di caricare, lanciare e tracciare un qualunque corso AICC compliant. Queste linee guida, soggette comunque a margini di interpretazione, sono state inizialmente sviluppate per l'industria dell'aviazione e si sono nel tempo diffuse fino a diventare degli standard di riferimento riconosciuti a livello internazionale.

¹⁴ Instructional Management Systems (IMS) è un consorzio tra partners aziendali, accademici e governativi: nato nel 1997 dalla EduCom National Learning Infrastructure, è oggi il punto di riferimento principale per la definizione delle specifiche riguardanti l'e-Learning.

¹⁵ Sharable Courseware Object Reference Model assieme ad AICC definiscono linee guida per lo sviluppo e la distribuzione di corsi on-line.

I metadata devono contenere il tipo di oggetto (testo, audio, immagine, etc.), l'autore, il proprietario dell'oggetto, i termini di distribuzione ed il formato.

Un esempio di cosa deve contenere un metadata, più completo è il seguente:

Nome – Etichetta assegnata al dato. *Identificatore* – Identificativo univoco assegnato al dato. *Versione* – Versione del dato. *Registrazione di autorità* – Entità autorizzata a registrare il dato. *Lingua* – Lingua nella quale il dato è indicato. *Definizione* – Indicazione che rappresenta chiaramente il concetto e la natura essenziale del dato. *Obbligatorietà* – Indica se il dato è richiesto sempre o solo in alcuni casi (contiene un valore). *Tipo di dato* – Indica la tipologia del dato che può essere rappresentato nel valore del dato stesso. *Occorrenza massima* – Indica un limite alla ripetibilità del dato. *Commento* – Un'osservazione che concerne l'applicazione del dato.

Anna Maria Tammaro docente all'università di Parma, afferma, in un articolo del marzo 2004, reperibile su elearningtouch.it che: «**Il Dublin Core** è lo standard emergente per la descrizione delle risorse Internet; [...] è uno schema abbastanza semplice ed è costituito solo da 15 elementi descrittivi, nessuno obbligatorio»¹⁶ come mostrato nella figura 1.1.

Nella figura vengono mappati (confrontati) gli elementi del Dublin Core con quelli descrittivi dell'IEEE/LOM.

Lo schema Dublin Core per le risorse didattiche (o Learning Objects) è accessibile a: <http://uk.dublincore.org/documents/education-namespace>

DC.Identifier	1.3:General.CatalogEntry. 1.1:General.Identifier is currently a reserved term, as there is no specified method for the creation of a globally unique identifier.
DC.Title	1.2:General.Title
DC.Language	1.4:General.Language
DC.Description	1.5:General.Description
DC.Subject	1.6:General.Keywords or 9:Classification with 9.1:Classification.Purpose equals "Discipline" or "Idea".
DC.Coverage	1.7:General.Coverage
DC.Type	5.2:Educational.LearningResourceType
DC.Date	2.3.3:LifeCycle.Contribute.Date when 2.3.1:LifeCycle.Contribute.Role has a value of "Publisher".
DC.Creator	2.3.2:LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role has a value of "Author".
DC.OtherContributor or	2.3.2:LifeCycle.Contribute.Entity with the type of contribution specified in 2.3.1:LifeCycle.Contribute.Role.
DC.Publisher	2.3.2:LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role has a value of "Publisher".
DC.Format	4.1:Technical.Format
DC.Rights	6:Rights
DC.Relation	7:Relation
DC.Source	7.2:Relation.Resource when the value of 7.1:Relation.Kind is "IsBasedOn".

Fig. 1.1 - (Elementi dello standard Dublin Core)

¹⁶ Art. - *Metadata per le risorse didattiche: una breve nota*, Dublin Core reperibile al link: <http://uk.dublincore.org/documents/education-namespace>.

Gli elementi dello standard Dublin Core sono i seguenti: (Identificatore, Titolo, Linguaggio, Descrizione, Soggetto e Parole Chiave, Copertura, Tipo di Risorsa, Data, Creatore, Altri autori di contributo subordinato, Editore, Formato dei dati, Gestione dei diritti, Relazione, Fonte.).

Ogni elemento del Dublin Core è definito utilizzando un insieme di dieci attributi dello standard ISO/IEC 11179 [ISO11179] per la descrizione dei dati. Questi includono: • **Nome** – Etichetta assegnata al dato; • **Identificatore** – Identificativo univoco assegnato al dato; • **Versione** – Versione del dato; • **Registrazione di autorità** – Entità autorizzata a registrare il dato; • **Lingua** – Lingua nella quale il dato è indicato; • **Definizione** – Indicazione che rappresenta chiaramente il concetto e la natura essenziale del dato; • **Obbligatorietà** – Indica se il dato è richiesto sempre o solo in alcuni casi (contiene un valore); • **Tipo di dato** – Indica la tipologia del dato che può essere rappresentato nel valore del dato stesso; • **Occorrenza massima** – Indica un limite alla ripetibilità del dato; • **Commento** – Un’osservazione che concerne l’applicazione del dato.

Fortunatamente sei, dei dieci, attributi sopra citati sono comuni a tutti gli elementi del DC. Questi sono, con i loro rispettivi valori: Versione 1.1. Registrazione di autorità Dublin Core Metadata Initiative. Lingua en. Obbligatorietà opzionale. Tipo di dato stringa di caratteri. Occorrenza massima illimitata .

Il set di metadati, attualmente, più noto è quello del progetto Instructional Management Systems **IMS** pubblicato nel 1998.¹⁷ I **LOM** sono scritti in **XML** (eXtensible Language Markup, definito come il formato universale per strutturare documenti e dati sul Web pensato per superare i limiti dell’HTML HyperText Markup Language) o nelle sue evoluzioni¹⁸, linguaggio di marcatura che aggiunge semantica alle pagine web; questo implica che si può dare, quindi, ai learning objects (che possono essere anch’essi scritti in XML) una caratterizzazione semantica molto importante per la loro reperibilità e riusabilità da parte di macchine automatiche. «XML consente a chiunque di progettare il proprio formato dei documenti e di scrivere poi in quel formato; inoltre è “*machine-readable*”(grazie al markup): i programmi possono leggere e “capire” il documento¹⁹».

¹⁷ Per l’elenco del set di metadati di IMS vedere articolo di Paola Perino del 26 maggio 2003 al link: http://www.blucomfort.com/internetime/tutto/pag_articolo.php?articolo_ID=30.

¹⁸ Per maggiore chiarezza si veda il capitolo 6

¹⁹ Laura Farinetti, Fulvio Corno, *L’Architettura del Semantic Web* Dip. Automatica e Informatica Politecnico di Torino, <http://elite.polito.it/courses/IIIliv/>

Utilizzando il linguaggio XML si possono etichettare parti di un testo con dei marcatori o tag; con `<indirizzo>via Salaria 113</indirizzo>` si cataloga il testo **via Salaria 113** come indirizzo, ma ciò per un calcolatore può risultare ambiguo in quanto indirizzo non viene sempre identificato in modo univoco, esistendo l'indirizzo di residenza, l'indirizzo e-mail, etc...

Per eliminare tale ambiguità la semantica associata ai metadati deve essere interpretata da chiunque nel medesimo modo, perciò i metadati debbono essere scritti in altri linguaggi, evoluzione dell'XML, che riescano a risolvere tali tipi di problematiche.

1.3 Generalità sulla progettazione di un Corso ODL

Costruire un generico corso in ODL che risponda ai criteri qualitativi odierni significa avere presente i requisiti essenziali che devono avere le quattro dimensioni dell'e-Learning: *Tecnologie, Servizi, Contenuti e Consulenza*.

Per le Tecnologie il processo didattico deve essere sostenuto da piattaforme online (**LCMS**) Learning Content Management System, evoluzione degli **LMS**, che siano possibilmente compatibili con i più famosi standard internazionali (AICC, SCORM, etc..²⁰).

Gli LCMS sono software che permettono di gestire in modo integrato e più efficiente le due principali variabili di un processo formativo: le *persone* ed i *contenuti*.

Un esempio di piattaforma LCMS è **IWT** (Intelligent Web Teacher)²¹ una piattaforma innovativa per la didattica intelligente su web.

Per detta degli autori «è stata realizzata con l'intento preciso di gettare le basi per l'e-Learning di futura generazione, infatti personalizza l'apprendimento sulle reali esigenze e preferenze dell'utente e garantisce estensibilità non solo a livello dei contenuti ma anche delle funzionalità soprattutto, a livello più alto, nelle strategie e nei modelli». Matteo Gaeta direttore del CRMPA²² sostiene che: «le caratteristiche innovative di IWT rispetto alle altre soluzioni di e-learning attualmente in commercio sono: possibilità di generazione automatica o assistita dei percorsi didattici a partire dagli obiettivi di apprendimento; possibilità di personalizzazione automatica dei corsi sulla base delle conoscenze pregresse dei singoli discenti e delle loro preferenze di apprendimento; supporto al monitoraggio ed alla valutazione automatica dei discenti sia in relazione alle conoscenze acquisite che alle abilità cognitive e capacità percettive mostrate; possibilità di estendere la piattaforma tramite Plug-In che consente l'aggiunta di servizi e tramite Driver che consentono la gestione di nuove tipologie e formati di contenuto».²³

²⁰ Vedi note precedenti

²¹ IWT, progettata da Nicola Capuano, Matteo Gaeta e Alessandra Micarelli , vedi all'indirizzo: http://www.crimpa.it/intraserv/documents/AIxA_2003.pdf

²² Centro di Ricerca di Matematica Pura ed Applicata

²³ IWT, Si afferma in Europa una piattaforma rivoluzionaria, Matteo Gaeta Direttore del CRMPA, vedi link: http://www.comunedasa.it/foram/topic.asp?TOPIC_ID=197

In generale, la risoluzione di un problema si compone di tre fasi teoriche consistenti in uno **stato iniziale** che necessita modificare, uno **stato finale** desiderato risoluzione del problema ed un insieme di **processi intermedi**, tra lo stato iniziale e finale, necessari alla risoluzione del problema.

Similmente si può risolvere il problema “Progetto dei contenuti di un corso di formazione in modalità e-Learning”.

Nello schema seguente **SIK** è lo stato di conoscenza iniziale posseduta dai partecipanti al corso, **TCK** è lo stato obiettivo di conoscenza finale che il corso si prefigge di trasmettere ai partecipanti, **TSIK** sono l’insieme dei passi (“stati”) intermedi necessari per modificare lo stato **SIK** al fine di raggiungere lo stato **TCK**.

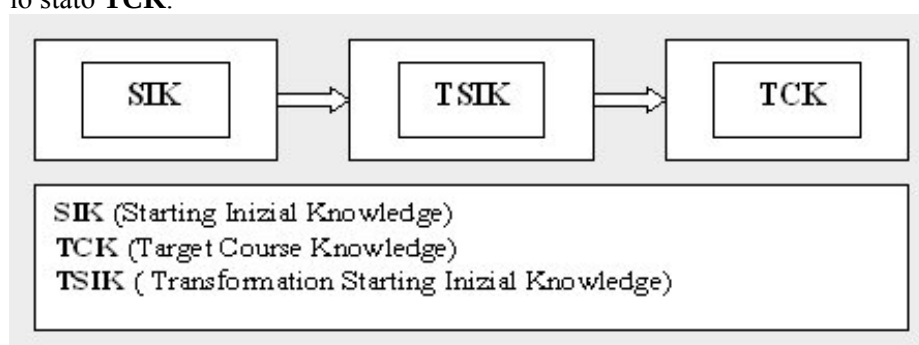


Fig. 1.2 - (Percorso di conoscenza)

Notiamo come questo schema in figura 1.2 sia molto generale la cui esplicitazione conduce a diverse metodologie di progettazione.

La progettazione, non soltanto dei contenuti ma, del corso di formazione in modalità e-Learning nella sua interezza richiede la esplicitazione delle *quattro fasi di cui si compone l’attività di soluzione di un problema*. (vedi fig. 1.3).

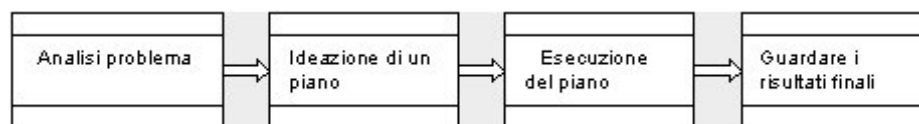


Fig. 1.3 - (Fasi di soluzione di un problema)

Oppure similmente, come in altri contesti, in quattro fasi **a)** definizione delle specifiche; **b)** progettazione; **c)** sviluppo; **d)** test.

L’attività di progettazione di un corso di formazione in modalità eLearning si può concepire come attività di risoluzione di un problema associandole come mostrato nella tabella 1.1 seguente.²⁴

²⁴ *E-Learning: una guida operativa*, G. Esposito, G. Maltese, ed. FrancoAngeli, Milano (2003) pp. 53-55

Tab. 1.1 - Risoluzione problema e progettazione corso

Fasi di risoluzione di un problema	Azioni per la progettazione corso
FASE DI ANALISI E COMPrensIONE DEL PROBLEMA	Questioni preliminari che il progettista si deve porre
	Perché c'è bisogno di questo corso? A chi serve questo corso? Che cosa si vuole trasmettere? Quante risorse si intendono impegnare nel progetto? Quale metodo didattico si vuole utilizzare?
	Raccolta di informazioni
	Analisi dei bisogni formativi; analisi dei destinatari generale; analisi dei destinatari specifica (<i>user profile</i>); analisi delle risorse necessarie per sviluppare il corso ed elencazione delle stesse (<i>resource listing</i> - persone, strumenti, finanziamenti, tempi...).
IDEAZIONE DI UN PIANO DI AZIONE	Pianificazione generale
	Definizione degli scopi del corso (o obiettivo generale); definizione degli obiettivi del corso (<i>learning objective</i>); definizione dei sotto-obiettivi formativi; strutturazione degli obiettivi.
	Pianificazione della didattica
	Scelta del metodo o modello didattico (<i>instructional approach</i>); scelta dei contenuti da erogare; definire il calendario delle attività corsali.
	Organizzare la comunicazione
	Analisi del ruolo del docente dei tutors e dei discenti; definizione di vari profili utente.
	Pianificazione della valutazione e del monitoraggio
	Predisposizione degli strumenti per il monitoraggio e la valutazione del progetto.
ESECUZIONE DEL PIANO	Costruzione ed erogazione del corso
GUARDARE I RISULTATI FINALI RAGGIUNTI	Valutazione e monitoraggio
	Osservazione dei risultati di monitoraggio e di valutazione del progetto.

1.3.1 Una metodologia di progetto di un generico corso

Attualmente lo scenario dell'e-Learning è caratterizzato dalla carenza di linee guida condivise e consolidate, anche se innumerevoli lavori sono stati prodotti; questo anche per la difficoltà intrinseca, che il tema pone, di creare standard qualitativi adatti ad ogni segmento della domanda di e-Learning²⁵.

Il nostro metodo di progettazione di un corso ODL di qualità, richiede un piccolo sforzo nel tener presente che esso sarà collocato all'interno di una certa Biblioteca la quale dovrà essere utilizzata tenendo in considerazione le dimensioni dell'e-Learning (*Servizi, Contenuti, Tecnologie e Consulenza*); quindi usando un approccio sistemico del tipo *top down*.

Nel progetto di un corso online pensiamo alla suddivisione dei contenuti delle lezioni in Learning Objects; questi si collocano al più basso livello gerarchico di una nostra ipotetica scala che preveda la suddivisione di una Biblioteca in Cataloghi, Corsi, Moduli, SottoModuli e Learning Objects.

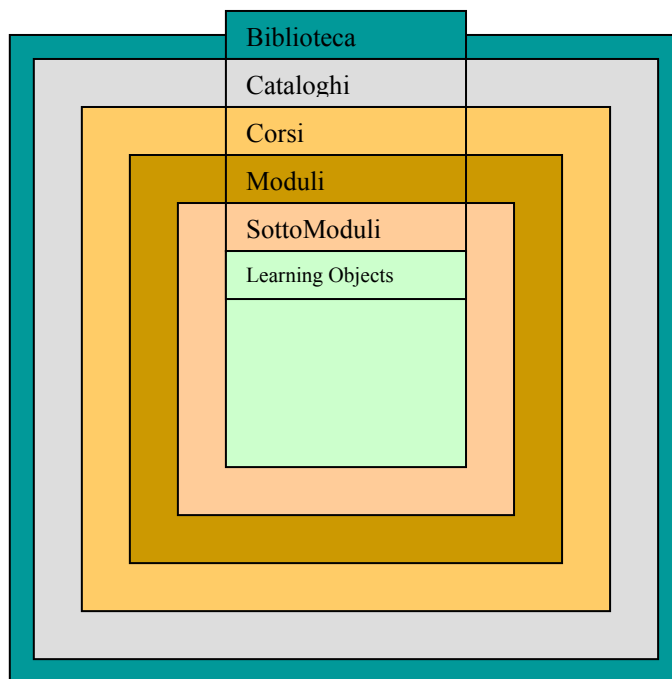


Fig. 1.4 - (Livelli di una Biblioteca)

²⁵ Criteri per la qualità nell'e-Learning, Ezio Fregnan e Federica Garbolino, ISVOR FIAT 2002, http://www.isvor.it/linee_guida.pdf

Al fine di stabilire il modello del corso introduciamo le seguenti definizioni.

Una Biblioteca (che assimiliamo ad una facoltà universitaria) è un insieme di Cataloghi.

- a) Un **Catalogo** (che assimiliamo ad un *corso di laurea* della facoltà universitaria) è un insieme di Corsi.
- b) Un **Corso** (che assimiliamo come un generico *titolo di un corso di laurea*) è un insieme di Moduli.
- c) Un **Modulo** è un insieme di SottoModuli, trattanti una omogenea tematica didattica.
- d) Un **SottoModulo** (che può essere o una lezione del corso o un test di autovalutazione o una Prova Intermedia) è un insieme di Learning Objects.

Precisiamo sin da subito che, nella pratica della costruzione dei corsi online, dobbiamo considerare un particolare Learning Object, il L.O. domanda/risposte. Si tratta di un learning object presente nei test di autovalutazione, che assumiamo essere la generica domanda e tutte le sue possibili opzioni risposta.

Definiamo un test di autovalutazione come un SottoModulo contenente l'insieme di tutti i learning objects di tipo domanda/risposte in esso presenti.

Definiamo una Prova intermedia come un SottoModulo erogato alla fine dei SottoModuli compresi nel Modulo che la contiene.

Consideriamo la nostra biblioteca **B** come formata da N cataloghi. La formula in forma compatta per calcolare, quali cataloghi $catal$ sono presenti nella biblioteca **B**, è data dall'unione di tutti gli N cataloghi cioè:

$$B = \bigcup_{c=1}^N catal$$

(1.1)

L'indice, dei cataloghi, c assume i valori $1, 2, \dots, N$.

Possiamo adesso costruirci la formula che calcola quali corsi appartengono alla nostra biblioteca **B** supponendo che il catalogo 1 ha M_1 corsi, il catalogo 2 ha M_2 corsi,..., il generico catalogo c -esimo ha M_c corsi,..., ed il catalogo N -esimo ha M_N corsi (con $M_1, M_2, \dots, M_c, \dots, M_N$ valori fissati ed in generale diversi fra loro).

$$B = \bigcup_{j_1=1}^{M_1} \text{CORSO}_{1, j_1} + \bigcup_{j_2=1}^{M_2} \text{CORSO}_{2, j_2} + \dots + \bigcup_{j_c=1}^{M_c} \text{CORSO}_{c, j_c} + \dots + \bigcup_{j_N=1}^{M_N} \text{CORSO}_{N, j_N} \quad (1.2)$$

La formula (1.2) scritta in forma compatta diventa:

$$B = \bigcup_{c=1}^N \bigcup_{j_c=1}^{M_c} \text{corso}_{c, j_c} \quad (1.3)$$

La formula (1.3) rappresenta la stessa biblioteca **B** come unione di tutti i corsi contenuti in ogni suo catalogo.

In questo caso il numero totale di corsi o titoli è dato dalla formula seguente:

$$N_{\text{corsiTotale}} = \sum_{c=1}^N M_c = M_1 + M_2 + \dots + M_N \quad (1.4)$$

L'indice j dei corsi, come abbiamo potuto notare, è funzione dell'indice dei cataloghi c , cioè: $j = f(c)$ poiché in cataloghi diversi possiamo avere un diverso numero di corsi.

Abbiamo inoltre diviso ogni corso in un insieme di Moduli.

Il generico corso corso_{c, j_c} è diviso in M_{c, j_c} moduli, con c che assume i valori da 1 fino ad N ($c = 1 \dots N$) e $j_c = 1 \dots M_c$.

Indicando con $\text{Mod}_{c, j_c, m_{c, j_c}}$ il generico modulo del catalogo c del

corso j_c , avente indice m_{c, j_c} avremo che l'insieme dei moduli della nostra biblioteca saranno dati dalla formula (1.5) (tenendo conto che il corso, del catalogo $c=1$, di indice j_1 avrà M_{1j_1} moduli, il corso del catalogo $c=1$ di indice j_N avrà M_{1j_N} moduli e così di seguito fino al corso del catalogo N di indice j_N che avrà M_{Nj_N} moduli. Il generico corso assumerà quindi i seguenti

valori: $m_{c, j_c} = 1, 2, 3, \dots, M_{c, j_c}$),

la biblioteca B è espressa come unione dei Moduli, sarà data dalla seguente formula:

$$B = \bigcup_{c=1}^N \bigcup_{j_c=1}^{M_c} \bigcup_{m_{c,j_c}=1}^{M_{c,j_c}} Mod_{c,j_c,m_{c,j_c}} \quad (1.5)$$

Questa formula serve per avere una previsione di quali sono i Moduli contenuti nella Biblioteca, avendo in questo modo associato ad ogni Modulo un indice univoco (Moduli etichettati). Ad esempio il quarto Modulo del terzo catalogo appartenente al quinto corso è indicato con Mod3,5,4.

Notiamo inoltre che nel caso in questione l'indice m dei Moduli è funzione di c e di jc, cioè $m = f(c, jc)$, poiché in Corsi diversi si può avere un numero diverso di Moduli.

Noto il numero di moduli di cui è composto ogni corso, il numero totale dei moduli usati nella nostra biblioteca B, **Nmodtotale**, si ricava applicando la seguente formula:

$$N_{modtotale} = \sum_{c=1}^N \sum_{j_c=1}^{M_c} M_{c,j_c} \quad (1.6)$$

Questa formula serve per avere una previsione di quanti sono in totale i Moduli usati nella Biblioteca e quindi poter fare tutte le analisi quantitative del caso. In particolare a noi servirà per capire quanti moduli abbiamo erogato nel nostro corso ODL (una misura della quantità di materiale sottoposta allo studente).

Ogni modulo di un generico corso appartenente ad un generico catalogo della nostra biblioteca può essere ulteriormente diviso in SottoModuli.

Chiamando l l'indice dei SottoModuli, indichiamo il generico

$$SMod_{c,j_c,m_{c,j_c},l_{c,j_c,m_{c,j_c}}}$$

SottoModulo con

e notando che in questo caso l è funzione di c, di jc e di mcjc cioè $l=f(c,jc,mcjc)$, avremo che l'insieme dei SottoModuli che compongono la nostra Biblioteca è dato dalla formula (1.7) seguente.

$$B = \bigcup_{c=1}^N \bigcup_{j_c=1}^{M_c} \bigcup_{m_{c j_c}=1}^{M_{c j_c}} \bigcup_{l_{c j_c m_{c j_c}}=1}^{L_{c j_c m_{c j_c}}} SMod_{c, j_c, m_{c j_c}, l_{c j_c m_{c j_c}}} \quad (1.7)$$

Il numero totale di SottoModuli, $SMOD_{totale}$, sarà dato dalla sommatoria (1.8) seguente.

$$SMOD_{totale} = \sum_{c=1}^N \sum_{j_c=1}^{M_c} \sum_{m_{c j_c}=1}^{M_{c j_c}} L_{c j_c m_{c j_c}} \quad (1.8)$$

$L_{c j_c m_{c j_c}}$ è il numero di SottoModuli formanti il generico Modulo $Mod_{c, j_c, m_{c j_c}}$.

Ultima formula, di questo discorso sequenziale, che ricaveremo è quella della Biblioteca B come unione dell'insieme di tutti i Learning Objects di cui si compone ciascun SottoModulo.

Indichiamo il generico Learning Object come nella seguente (1.9)

$$LOB_{c, j_c, m_{c j_c}, l_{c j_c m_{c j_c}}, o_{c j_c m_{c j_c} l_{c j_c m_{c j_c}}}} \quad (1.9)$$

Indicando con

$$LO_{c j_c m_{c j_c} l_{c j_c m_{c j_c}}}$$

il numero di learning objects, (presenti nel generico sottomodulo l , del generico catalogo, del generico corso, del generico modulo), avremo la formula cercata la (1.10) che dà la biblioteca B come unione di tutti questi learning object **LOB**.

$$B = \bigcup_{c=1}^N \bigcup_{j_c=1}^{M_c} \bigcup_{m_{c,j_c}=1}^{M_{c,j_c}} \bigcup_{l_{c,j_c,m_{c,j_c}}=1}^{L_{c,j_c,m_{c,j_c}}} \bigcup_{o_{c,j_c,m_{c,j_c},l_{c,j_c,m_{c,j_c}}}=1}^{LO_{c,j_c,m_{c,j_c},l_{c,j_c,m_{c,j_c}}}} LOB_{c,j_c,m_{c,j_c},l_{c,j_c,m_{c,j_c}},o_{c,j_c,m_{c,j_c},l_{c,j_c,m_{c,j_c}}}}$$

(1.10)

Il numero totale di Learning Objects (usati nei corsi, anche eventualmente sotto nomi diversi) L.O.totale sarà dato dalla seguente sommatoria:

$$L.O.totale = \sum_{c=1}^N \sum_{j_c=1}^{M_c} \sum_{m_{c,j_c}=1}^{M_{c,j_c}} \sum_{l_{c,j_c,m_{c,j_c}}=1}^{L_{c,j_c,m_{c,j_c}}} LO_{c,j_c,m_{c,j_c},l_{c,j_c,m_{c,j_c}}}$$

26

(1.11)

Notiamo che nel caso in questione l'indice o dei learning objects è funzione di $c, j_c, m_{c,j_c}, l_{c,j_c,m_{c,j_c}}$ cioè $o = f(c, j_c, m_{c,j_c}, l_{c,j_c,m_{c,j_c}})$.

Consideriamo un esempio di applicazione delle suddette formule ad una Biblioteca composta da 3 cataloghi ($N=3$), i quali contengono 3 corsi il primo, 4 corsi il secondo e 2 corsi il terzo (vedi Tabella 1.2).

Il primo corso del catalogo 1 ha $M11=3$ moduli, il secondo corso ne ha $M12=2$, il terzo ne ha $M13=2$ e così di seguito per gli altri due cataloghi.

²⁶ Nel nostro modello ipotizziamo che i vari livelli di cui si compone la biblioteca siano disgiunti. Ad esempio se un Learning Object è presente in un sottomodulo i -esimo non può essere presente nel sottomodulo j -esimo oppure se esso è presente in entrambi i sottomoduli ha due etichette diverse "indici diversi".

Tab. 1.2 - Generica Biblioteca divisa in Moduli

Mod1,1,1	Mod1,1,2	Mod1,1,3	Mod2,1,1	Mod2,1,2	Mod3,1,1	Mod3,1,2	Mod3,1,3
Corso1,1 - M11=3			Corso2,1 - M21=2		Corso3,1 - M31=3		
Mod1,2 ,1	Mod1,2,2		Mod2,2,1	Mod2,2,2	Mod3,2,1	Mod3,2,2	Mod3,2,3
Corso1,2 - M12=2			Corso2,2 - M22=2		Corso3,2 - M32=3		
Mod1,3 ,1	Mod1,3 ,2		Mod2,3,1	Mod2,3,2			
Corso1,3 - M13=2			Corso2,3 - M23=2				
			Mod2,4,1	Mod2,4,2			
			Corso2,4 - M24=2				
.			.		.		
c=1			c=2		c=3		
Biblioteca B							

La formula (1.5) applicata a questo esempio dà la Biblioteca **B** come unione di tutti i moduli della Tabella 1.2, cioè

$$\begin{aligned}
 B = & \text{Mod1,1,1} \cup \text{Mod1,1,2} \cup \text{Mod1,1,3} \cup \text{Mod1,2,1} \cup \text{Mod1,2,2} \cup \text{Mod1,3,1} \cup \text{Mod1,3,2} \cup \\
 & \text{Mod2,1,1} \cup \text{Mod2,1,2} \cup \text{Mod2,2,1} \cup \text{Mod2,2,2} \cup \text{Mod2,3,1} \cup \text{Mod2,3,2} \cup \text{Mod2,4,1} \cup \text{Mod2,4,2} \cup \\
 & \text{Mod3,1,1} \cup \text{Mod3,1,2} \cup \text{Mod3,1,3} \cup \text{Mod3,2,1} \cup \text{Mod3,2,2} \cup \text{Mod3,2,3}
 \end{aligned}$$

Il numero totale di moduli di questa Biblioteca applicando la formula (1.6):

$$N_{mod\ totale} = \sum_{c=1}^N \sum_{j_c=1}^{M_c} M_{cj_c} \quad \text{è:}$$

$$N_{mod\ totale} = (M_{11} + M_{12} + M_{13}) + (M_{21} + M_{22} + M_{23} + M_{24}) + (M_{31} + M_{32}) = (3 + 2 + 2) + (2 + 2 + 2 + 2) + (3 + 3) = 21.$$

1.3.2 Metodologia didattica di assegnazione delle prove intermedie

Una Prova Intermedia generica $PI_{c, j_c, m_c j_c}$ è anch'essa un **SottoModulo** che è erogato al learner subito dopo lo studio del restante

corrispondente modulo $Mod_{c, j_c, m_c j_c}$ cui essa appartiene.

Una metodologia didattica di assegnazione delle prove intermedie, le quali hanno almeno tre principali motivi d'essere presenti, in un corso a distanza sono:

- a) Verifica, da parte dei tutors, mentors e docenti, dell'apprendimento

acquisito dai learners studiando il modulo $Mod_{c, j_c, m_c j_c}$

- b) Diminuzione della sensazione d'isolamento, insita in un corso ODL, dei learners, (in quanto il cercare di affrontare positivamente le prove intermedie, spinge il discente a relazionarsi maggiormente con gli altri protagonisti dell'aula virtuale, presente nella piattaforma LCMS, colleghi di corso, docenti e tutors "apprendimento collaborativo").

- c) Strumento per attribuire i crediti formativi intermedi, prevede il seguente schema, consequenziale.

Indichiamo con $NPI_{c, j_c, m_c j_c}$ il Numero di Prove di valutazione

dell'apprendimento Intermedie presenti nel $Mod_{c, j_c, m_c j_c}$.

Il nostro metodo didattico prevede $NPI_{c, j_c, m_c j_c} = 1$ ossia l'assegnazione di una sola prova intermedia per ogni modulo presente nel

$CORSO_{c, j_c}$. Reindicando con M_{c, j_c} il numero di moduli presenti nel generico $CORSO_{c, j_c}$, in generale si avrà che il numero totale di prove intermedie presenti in tale corso sono:

$$N_{totalePI_{c, j_c}} = M_{c, j_c} * NPI_{c, j_c, m_{c, j_c}} \quad 27$$

(1.12)

mentre il numero totale di prove intermedie **NPIbit** all'interno della biblioteca B si può ricavare come mostrato nella 1.13.

$$NPIbit = \sum_{c=1}^N \sum_{j_c=1}^{M_c} \sum_{m_{c, j_c}=1}^{M_{c, j_c}} NPI_{c, j_c, m_{c, j_c}} \quad (1.13)$$

Supposto di avere una sola **P.I.** per ogni Modulo, essendo le Prove Intermedie del corso erogate alla fine del penultimo SottoModulo di un certo Modulo, secondo la simbologia sin qui usata, avranno la seguente etichetta:

$$SMod_{c, j_c, m_{c, j_c}}, L_{c, j_c, m_{c, j_c}} \quad \text{dove con } L_{c, j_c, m_{c, j_c}} \text{ indichiamo al solito}$$

il numero totale di SottoModuli contenuti nel generico Modulo m_{c, j_c} .

Usando l'esempio della Biblioteca mostrata in Tabella 1.2 se siamo in Mod3,1,2, ossia nel secondo Modulo del primo Corso del terzo Catalogo e Mod3,1,2 contiene 7 SottoModuli cioè $L_{312} = 7$, avremo che la Prova Intermedia presente in tale modulo è indicata ed etichettata con SMod3,1,2,7.

²⁷ Nel seguito mostreremo che il nostro corso ODL ha un primo modulo ed un ultimo modulo costituiti da un unico SottoModulo, rispettivamente, Test Introduttivo e Prova Finale privi logicamente di Prove intermedie, per cui in tal caso nella 1.12 M_{c, j_c} è decrementato di 2.

1.3.3 Metodologia di assegnazione dei crediti formativi nelle prove intermedie

I crediti formativi hanno una funzione molto importante in un corso ODL. Attraverso essi si premia parzialmente il lavoro svolto dal discente nelle fasi intermedie del corso; inoltre servono da stimolo per il learner e da autoverifica del livello di apprendimento raggiunto, prima di affrontare la prova finale (esame), trattante argomenti di tutto il corso.

E' importante introdurre una metodologia di assegnazione dei crediti formativi che risponda a criteri di equità e che sia abbastanza in sintonia con le attese dei learners.

La metodologia che noi introduciamo è la seguente: si predispone nel generico $corso_{c, j_c}$ una prova di valutazione intermedia $PI_{c, j_c, m_{c, j_c}}$

per ogni $Mod_{c, j_c, m_{c, j_c}}$ che deve eseguirsi dopo lo studio del modulo, corrispondente restante, da parte del learner.

L'esito di queste prove intermedie, ricavato in automatico da software dei LCMS, è valutato in trentesimi e reso disponibile allo studente, assieme al credito formativo da essa prodotto, dentro la scheda utente della piattaforma LCMS, per dare al learner visione e consapevolezza del suo rating generale.

Il credito formativo, prodotto, è ricavato dal voto riportato nella prova intermedia nel seguente modo, dopo aver introdotto due definizioni.

Definizione a)

Si definisce un *credito formativo totale massimo*, il punteggio **Maxcredito**, che si raggiunge soltanto nel caso di svolgimento ottimale da parte del learner di tutte le Prove Intermedie di un corso; noi sceglieremo un

ad esempio il valore $Maxcredito_{c, j_c} = 6 \text{ punti}$;

Definizione b)

Si definisce *Peso della Prova Intermedia* **PesoPI** il valore numerico che si assegna ad ogni prova intermedia; ossia si assegna ad ogni prova intermedia

un peso $PesoPI_{c, j_c, m_{c, j_c}}$, che può assumere ad es. il valore 1, 2 o 3 (attribuito in base al criterio della difficoltà di apprendimento dell'argomento oggetto della prova).

Introduciamo l'indice k per scandire tutte le prove intermedie di un corso, l'indice s per scandire le prove intermedie svolte dal learner e l'indice n per scandire le prove intermedie non svolte dal learner; si definisce Voto Prove

Intermedie pesato della prova s -esima $VPIp_{c,j_c,m_c,j_c^s}$

il prodotto fra il $PesoPI_{c,j_c,m_c,j_c^s}$

della s -esima prova intermedia ed il voto riportato dal learner in tale prova, normalizzato a 30, in formula:

$$VPIp_{c,j_c,m_c,j_c^s} = PesoPI_{c,j_c,m_c,j_c^s} * Voto_s / 30 \quad (1.14)$$

Detto, NPI svolte, il numero di prove intermedie affrontate dal learner, il Voto pesato delle prove intermedie positive affrontate è dato dalla formula seguente: ²⁸

$$PesoVotoPI^+_{c,j_c} = \frac{\sum_{s=1}^{NPI\ svolte} VPIp_{c,j_c,m_c,j_c^s}}{\sum_{k=1}^{NPI\ totali} PesoPI_{c,j_c,m_c,j_c^k}} \quad (1.15)$$

Adesso possiamo calcolare il Credito Risultante come la parte intera superiore del prodotto fra il Voto pesato delle prove intermedie positive ed il Maxcredito di quel corso, in formula:

$$Credito_{c,j_cRisul} = \left\lceil PesoVotoPI^+_{c,j_c} * Maxcredito_{c,j_c} \right\rceil \quad (1.16)$$

²⁸ Notare che NPI_{totali} è lo stesso termine $N_{totalePIc,jc}$ ricavato nella (1.12)

Detto, $NPI\ nonsvolte$, il numero di prove intermedie non affrontate dal learner, il Voto pesato delle prove intermedie negative (non affrontate) è dato dalla formula 1.17.

$$PesoVotoPI^{-}c, j_c = \frac{\sum_{n=1}^{NPI\ nonsvolte} PesoPI_{c, j_c, m_c j_c}^n}{\sum_{k=1}^{NPI\ totali} PesoPI_{c, j_c, m_c j_c}^k}$$

(1.17)

Notare che: sia il Voto pesato delle prove intermedie positive è ≤ 1 sia il Voto pesato delle prove intermedie negative è ≤ 1 .

Assumiamo inoltre la convenzione che, se il voto riportato in una prova intermedia è inferiore a 18 la prova si assume non essere stata affrontata (con conseguente creazione ed accumulo di un Debito formativo).

A questo punto calcoliamo come *Debito formativo Risultante* la parte intera superiore (presa col segno meno) del prodotto fra il Voto pesato delle prove intermedie negative ed il Maxcredito di quel corso, in formula:

$$Debito_{c, j_cRisul} = - \left[PesoVotoPI^{-}c, j_c * Maxcredito_{c j_c} \right]$$

(1.18)

Finalmente siamo giunti alla formula del credito formativo che è assegnato allo studente quando si cimenta in tutte od alcune prove intermedie del $CORSO_{c, j_c}$.

$$Credito_{assegnato\ c, j_c} = Credito_{c, j_cRisul} + Debito_{c, j_cRisul}$$

(1.19)

Il valore ricavato applicando la (1.19) al termine del corso si sommerà al voto riportato dallo studente nel test di valutazione finale, per dare origine al **Voto Esame**. Consideriamo un esempio chiarificatore di calcolo dei crediti

formativi utilizzando la teoria di questo paragrafo. Supponiamo che un learner in un corso composto da sette moduli debba affrontare sette prove di valutazione intermedie (una prova per ogni modulo). Nella Tabella 1.3 riportiamo i voti conseguiti dallo studente nelle cinque prove intermedie affrontate ed il peso assegnato a ciascuna prova.

Tab. 1.3 - Esempio di Voti e Pesì Prove Intermedie

	Prova 1	Prova 2	Prova 3	Prova 4	Prova 5	Prova 6	Prova 7
Voto	24	-	20	18	30	-	26
Peso	1	2	3	2	2	1	3

$$VPIp_{c,j_c,m_c j_c s} = PesoPI_{c,j_c,m_c j_c s} * Voto_s / 30$$

$$PesoVotoPI^+_{c,j_c} = \frac{\sum_{s=1}^{NPI\ svolte} VPIp_{c,j_c,m_c j_c s}}{\sum_{k=1}^{NPI\ totali} PesoPI_{c,j_c,m_c j_c k}}$$

$$VPIp = 24*1/30 + 20*3/30 + 18*2/30 + 30*2/30 + 26*3/30 \\ = 0,8 + 2 + 1,2 + 2 + 2,6 = 8,6$$

$$PesoVotoPI^+_{c,j_c} = 8,6 / 14 = 0,781 \text{ che è } \leq 1$$

$$Credito_{c,j_cRisul} = \left\lceil PesoVotoPI^+_{c,j_c} * Maxcredito_{c,j_c} \right\rceil$$

$$\text{Il Credito Risultante} = \left\lceil 0,781 * 6 \right\rceil = \left\lceil 4,69 \right\rceil = +5$$

$$PesoVotoPI^{-}_{c,j_c} = (2+1)/14 = 0,214 \text{ che è } \leq 1.$$

Il debito sarà applicando la seguente formula:

$$Debito_{c,j_c \text{Risul}} = - \left[PesoVotoPI^{-}_{c,j_c} * Maxcredito_{c,j_c} \right]$$

$$= \left[0,214 * 6 \right] = \left[1,284 \right] = -2$$

Il credito formativo di corso assegnato al learner in sede di valutazione finale sarà $= + 5 + (-2) = +3$.

Questo punteggio andrà a sommarsi al voto acquisito nel test finale per formare il *Voto esame*.

1.3.4 Applicazione delle formule al corso FIODL

Il corso a distanza, da noi progettato, FIODL²⁹ è reperibile al link: <http://digilander.libero.it/capano/index.html> è inserito in una sola biblioteca contenente un unico catalogo c=1 e due corsi (M1=2): corso per principianti *Corso1,1* e corso avanzato *Corso1,2*, predisposti rispettivamente per gli studenti di Fondamenti di Informatica 1 e per gli studenti di Fondamenti 2 del consorzio NET.T.UN.O³⁰.

Tab. 1.4 - Moduli e SottoModuli F1

Ripartizione Corso1,1 in Moduli e SottoModuli	
Moduli	SottoModuli
Mod1,1,1	(Test Introduttivo al corso Fondamenti 1) $\equiv$ (SMod1,1,1,1)
Mod1,1,2	(L1, T1, L2, T2, L3, T3, L4, T4, L5, T5)
	(Prova Intermedia 1 per studenti Fondamenti1)
Mod1,1,3	(L6, T6) = (SMod1,1,3,1 , SMod1,1,3,2)
	SMod1,1,3,3 $\equiv$ (Nessuna Prova intermedia)
Mod1,1,4	(L7, T7, L8, T8)
	(Prova intermedia 2 per Fondamenti 1)
Mod1,1,5	(L9, T9, L10, T10, L11, T11)
	(Prova Intermedia 3 per F1)
Mod1,1,6	(L12, T12)
	(Prova Intermedia 4 per F1)
Mod1,1,7	(L13, T13) $\equiv$ (SMod1,1,7,1 , SMod1,1,7,2)
	(Prova Intermedia 5 per F1) $\equiv$ (SMod1,1,7,3)
Mod1,1,8	(L14, T14, L15, T15) (SMod1,1,8,1 , SMod1,1,8,2)
	(Prova Intermedia 6 per F1)
Mod1,1,9	(L16, T16)
	(Prova Intermedia 7 per F1)
Mod1,1,10	(Prova Finale del corso Fondamenti 1)

²⁹ Fondamenti di Informatica in Open and Distance Learning

³⁰ **NET**work **T**elematico per l'**UN**iversità **O**vunque.

Il corso per principianti *Corso1,1* era composto da 10 Moduli comprendenti un totale di 41 SottoModuli (1 test introduttivo³¹, 16 lezioni, 16 test di autovalutazione, 7 Prove Intermedie ed 1 Prova Finale³²) (vedi Tabella 1.4).

Gli studenti di Fondamenti 1 dovevano seguire un percorso di 10 Moduli da Mod1,1,1 a Mod1,1,10. Il Modulo Mod1,1,1 è composto dai SottoModuli (Test Introduttivo per Fondamenti1).

Il Modulo Mod1,1,2 è composto dai SottoModuli L1, T1, L2, T2, L3, T3, L4, T4, L5, T5, prima Prova Intermedia di Valutazione per Fondamenti 1 e così di seguito per i restanti otto Moduli come mostrato nella Tabella 1.3.

L'aggregazione dei SottoModuli in Moduli è fatta rispettando alcune caratteristiche logiche. Ad esempio le prime cinque lezioni (L1, L2, L3, L4, L5 ed i rispettivi test di autovalutazione T1, T2, T3, T4, T5 del modulo Mod1,1,2), trattano tutte quante un unico argomento didattico in modo "completo" (Architettura dei calcolatori elettronici).

Ciascun SottoModulo di tipo lezione Li, a sua volta, è composto da un determinato numero di learning objects.

I SottoModuli, test di autovalutazione Ti, sono composti da un numero di learning objects pari al numero di domande presenti nel test.

Domanda e possibili risposte, come da definizione, nel SottoModulo di tipo test di autovalutazione, li consideriamo essere un unico Learning Object.

³¹ Il test introduttivo è considerato essere, nel nostro modello, il primo Modulo erogato nel corso cui esso appartiene.

³² La Prova Finale è considerata essere, nel nostro modello, l'ultimo Modulo erogato nel corso cui essa appartiene.

Il corso avanzato *Corso1,2* era composto da 9 Moduli comprendenti un totale di 27 SottoModuli (1 test introduttivo, 9 lezioni, 9 test di autovalutazione, 7 Prove Intermedie ed 1 test finale) (vedi tabella 1.5).

Tab. 1.5 - Moduli e SottoModuli F2

Ripartizione Corso1,2 in Moduli e SottoModuli	
Moduli	SottoModuli
Mod1,2,1	(Test Introduttivo al corso Fondamenti 2) $\equiv$ (SMod1,2,1,1)
Mod1,2,2	(L13, T13) (Prova Intermedia 1 per F2)
Mod1,2,3	(L14, T14, L15, T15) (Prova Intermedia 2 per F2)
Mod1,2,4	(L16, T16) $\equiv$ (SMod1,2,4,1 , SMod1,2,4,2) (Prova Intermedia 3 per F2) $\equiv$ (SMod1,2,4,3)
Mod1,2,5	(L17, T17) (Prova Intermedia 4 per F2)
Mod1,2,6	(L18, T18) (Prova Intermedia 5 per F2)
Mod1,2,7	(L19, T19) (Prova Intermedia 6 per F2)
Mod1,2,8	(L20, T20, L21, T21) (Prova Intermedia 7 per F2)
Mod1,2,9	(Prova finale del corso Fondamenti 2)

Per la Tabella 1.5 valgono considerazioni analoghe a quelle fatte per la Tabella 1.4.

Nel corso FIODL si aveva che le lezioni L13, L14, L15, L16 erano comuni ai due corsi come anche i corrispondenti Test di autovalutazione, mentre erano diverse le prove intermedie dei Moduli corrispondenti.

Nel nostro schema, a due corsi diversi, quindi corrisponde, per quanto riguarda i SottoModuli, uguale contenuto ma etichetta diversa. Precisamente per quanto riguarda la lezione 13 ed il test 13 abbiamo coincidenza fra il contenuto dei sottomoduli (SMod1,1,7,1 $\equiv$ SMod1,2,8,1) e (SMod1,1,7,2 $\equiv$ SMod1,2,8,2).

In questo tipo d'approccio possiamo vedere i SottoModuli come un aggregato di Learning Objects, che assolvono (al pari dei learning objects) ad un compito didattico prefissato, omogeneo e temporalmente sostenibile da parte dei discenti (*Obiettivo didattico del SottoModulo*).

Pensiamo che, da rilevazioni fatte da alcuni ricercatori emerge, dallo studio della curva di attenzione media di un discente medio, un valore che si assesta sui venti minuti³³, perciò si consiglia che la durata di un learning object non debba sicuramente superare tale tempo.

In termini didattici L16 $\cup$ T16 appare essere un “super-SottoModulo” con obiettivo formativo comune essendo T16 un test di autovalutazione dei contenuti della lezione L16. Ossia in termini di obiettivi didattici i vari componenti della nostra struttura sono tutti interrelati, cioè cooperano tutti quanti al raggiungimento del medesimo obiettivo formativo.

Nelle tabelle 1.4 e 1.5 abbiamo applicato la formula (1.5):

$$B = \bigcup_{c=1}^N \bigcup_{j_c=1}^{M_c} \bigcup_{m_{c,j_c}=1}^{M_{c,j_c}} Mod_{c,j_c,m_{c,j_c}}$$

con $c=1$ $j_1=1,2$ $M_{11}=10$ e $M_{12}=9$.

Quindi la biblioteca, in cui è contenuto il corso FIODL, è costituita dall'unione di 19 Moduli³⁴, quelli elencati in Tabella 1.4 e Tabella 1.5.

Adesso esplicitiamo, in Tabella 1.6, uno fra i 19 Moduli (Mod1,2,4) del Corso1,2 di FIODL, in funzione dei SottoModuli e dei corrispondenti Learning Objects, ricordandoci del fatto che il Test di autovalutazione 16 (T16) è costituito da 8 domande e relative opzioni risposta (con 4 scelte possibili, di cui una esatta, per ogni domanda), quindi 8 L.O.³⁵ (vedi appendice D) a cui associamo 8 L.O. appartenenti al SottoModulo di tipo Lezione (L16) didatticamente corrispondenti.

³³ Articolo di Federica Garbolino, di ISVOR FIAT, *Dalla Fad all'e-learning: la formazione a distanza non è più così distante*, pag. 20, al link: <http://www.isvor.it/e-learning1.html>

³⁴ 8 SottoModuli erano comuni ai due corsi, ma avendo prove intermedie diverse non possiamo dire che i corrispondenti Moduli sono comuni, per cui è vero che in totale i Moduli sono 19.

³⁵ Ricordiamo che, nella pratica della costruzione di corsi, consideriamo nei test di autovalutazione un tipo particolare di learning object: il L.O. domanda/risposte possibili

Tab. 1.6 - Divisione Modulo di F2 in SottoModuli e L.O.

Modulo Mod1,2,4		
SMod1,2,4,1≡(Lezione16)	SMod1,2,4,2≡ (Test autov. 16)	SMod1,2,4,3≡ (Prova Intermedia 3 per F2)
LO1,2,4,1,1	LO1,2,4,2,1	LO1,2,4,3,1
LO1,2,4,1,2	LO1,2,4,2,2	
LO1,2,4,1,3	LO1,2,4,2,3	
LO1,2,4,1,4	LO1,2,4,2,4	
LO1,2,4,1,5	LO1,2,4,2,5	
LO1,2,4,1,6	LO1,2,4,2,6	
LO1,2,4,1,7	LO1,2,4,2,7	
LO1,2,4,1,8	LO1,2,4,1,8	

Nell'appendice D, esplicitiamo i L.O. di **T16** e di **L16**, abbiamo associato ad ogni L.O. appartenente a T16 un “*corrispondente didatticamente*” L.O. appartenente a L16. Nella figura 1.5 mostriamo un generico esempio di costruzione/associazione dei 3 L.O. di tipo lezione a partire dai 3 L.O. di tipo test di autovalutazione.

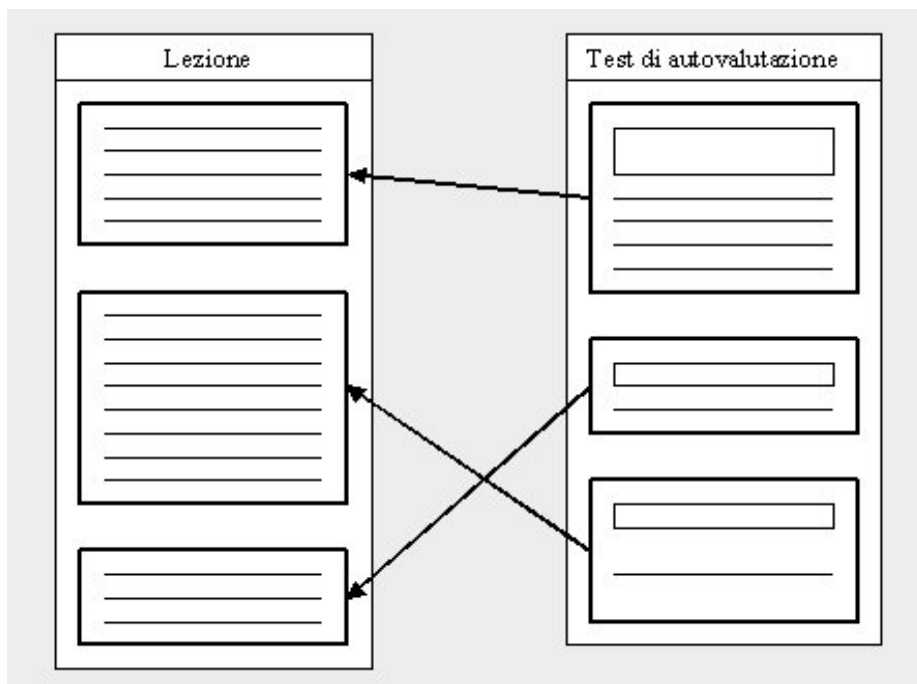


Fig. 1.5 - (Corrispondenza fra L.O. di tipo test e L.O. di tipo Lezione)

Il SottoModulo, Prova Intermedia 3³⁶, per i learners del Corso di Fondamenti di Informatica 2 (F2), che era da svolgersi dopo lo studio dei due SottoModuli (L16, T16) è riportata in figura 1.6.

Fig. 1.6 - (Prova Intermedia 3 per F2)³⁷

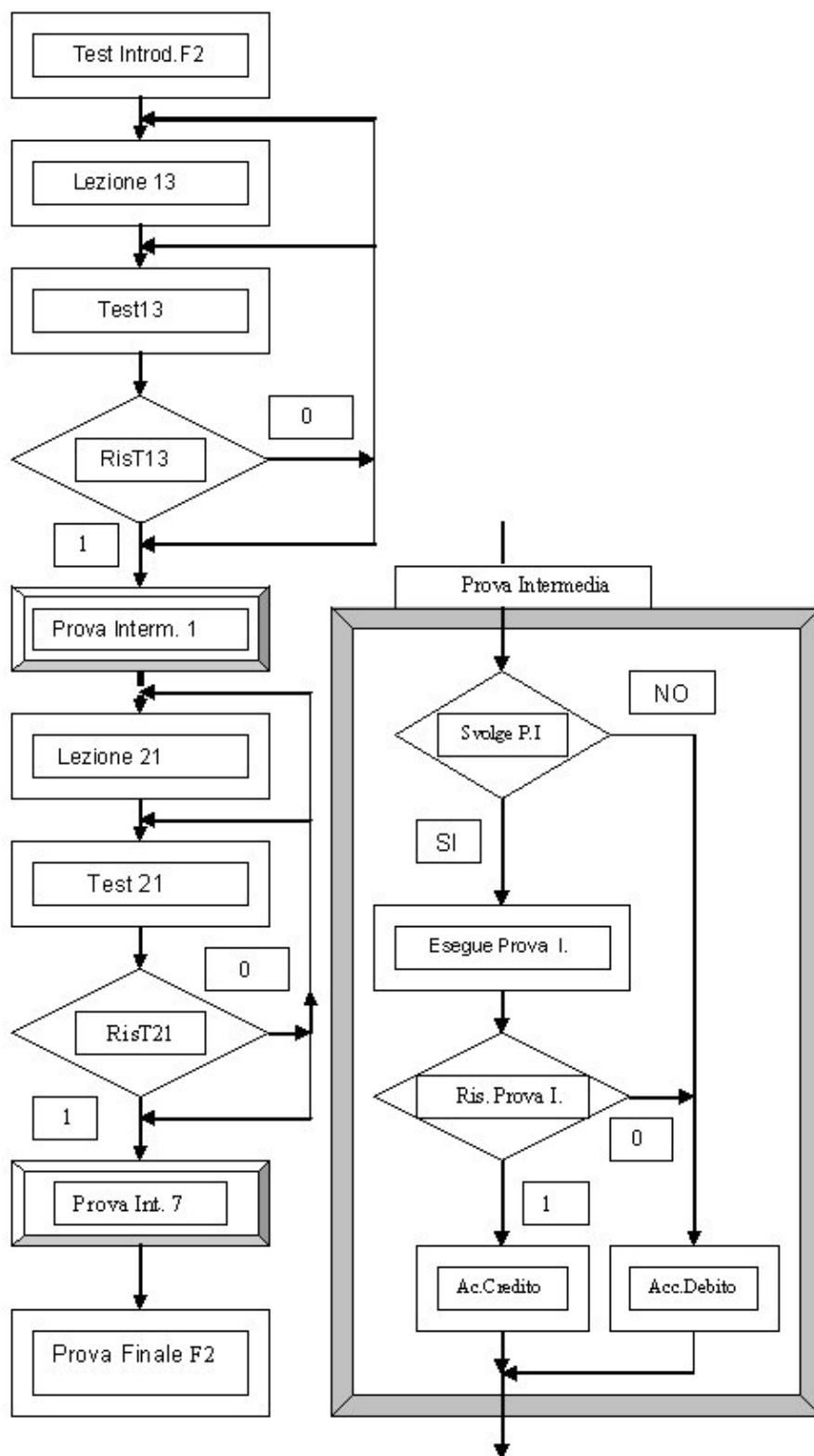
Per concludere questa sottosezione, schematizziamo attraverso un diagramma di flusso il percorso didattico più verosimile che, nel corso1,2, ha seguito un learner di Fondamenti 2.

Fig. 1.7 - (Diagramma di flusso seguito dal learner nel corso F2)

³⁶ La Prova Intermedia la consideriamo essere un *particolare SottoModulo costituito in genere da un unico learning object* del tipo (traccia prova)/(possibili soluzioni).

³⁷ Vedere l'appendice D per la soluzione proposta alla Prova Intermedia 3 (possibile soluzione).

Percorso didattico Learners F2



Illustriamo a parole il diagramma di figura 1.7. Il learner Si,2 eseguiva, come primo passo del corso^{1,2}, il Test Introduttivo (Particolare Modulo costituito da un unico SottoModulo) formato da 30 domande (30 learning objects); unico passo “obbligato” del corso.

Proseguiva il corso con lo studio della lezione 13 ed alla fine eseguiva il corrispondente Test di autovalutazione¹³; se il risultato numerico riportato in tale test 13 (RisT13) era insufficiente riandava o a ristiudiarsi il SottoModulo lezione 13 e rieseguiva il test 13 o rieseguiva soltanto il test 13 senza studiarsi la lezione (probabilmente adottando il motto “sbagliando si impara”) oppure passava ad eseguire la prima Prova Intermedia.

Se il risultato, riportato nel test 13, era positivo eseguiva la prima Prova Intermedia (esercitazione di fine Modulo).

Questo processo è ripetuto con la L14, T14 etc. fino alla lezione 21, al test 21 ed alla ultima Prova Intermedia 7. (Nello corso F2, abbiamo 9 lezioni, 9 test di autovalutazione e 7 Intermedie; ciò è dovuto al fatto che si è assegnata una P.I. per ogni Modulo che in generale contiene un numero maggiore uguale a due di lezioni e test di autovalutazione).

Nel diagramma strutturato mostrato nel precedente diagramma di flusso corrispondente alla generica Prova Intermedia abbiamo che, se il learner non svolgeva la Prova “oppure la svolgeva con voto insufficiente” accumulava un debito formativo altrimenti accumulava un credito formativo.

Come ultimo passo, nel diagramma, vi è la Prova Finale il cui risultato numerico, riportato dal learner, (dopo correzione e valutazione della Prova da parte del tutor) si sommava al *credito formativo risultante totale*, dalle varie Prove Intermedie, per dare luogo al *Voto Esame*.

1.3.5 Obiettivi formativi

Compreso che nell'e-Learning il focus deve essere posto sulle soluzioni di apprendimento, possiamo replicare ed adattare, lo schema teorico a livelli della figura. 1.4 (I cinque livelli di una biblioteca), agli obiettivi formativi che si deve prefiggere una facoltà universitaria on-line (biblioteca B), un suo corso di laurea (catalc), un corso all'interno del corso di laurea c ($corso_{c, j_c}$), un insieme di argomenti omogenei di quel corso j_c

($Mod_{c, j_c, m_{c j_c}}$), le lezioni, i test, le prove intermedie e quant'altro necessario ad esaudire in forma qualitativa ed esaustiva gli argomenti

$$SMod_{c, j_c, m_{c j_c}, l_{c j_c, m_{c j_c}}}$$

omogenei di quel corso ($l_{c j_c, m_{c j_c}}$) e le unità didattiche che compongono i SottoModuli cioè i learning objects LOB

$$LOB_{c, j_c, m_{c j_c}, l_{c j_c, m_{c j_c}}, o_{c j_c, m_{c j_c}, l_{c j_c, m_{c j_c}}}}.$$

Notiamo che se si introducesse nella figura 1.4 un ulteriore livello, che contenga l'insieme di tutte le biblioteche B_i con ($i=1 \dots N_{Facoltà}$), chiamato ad esempio "SuperBiblioteca" avremmo lo schema teorico completo per rappresentare una certa Università e così di seguito per rappresentare le tutte le Università italiane (ITBiblioteca etc..).

Ogni livello, del nostro schema, deve avere degli obiettivi formativi; ad esempio uno fra gli obiettivi del livello Cataloghi è di permettere che in base alla selezione, ed al superamento in sede di esame da parte del learner di un insieme di titoli "es. quelli compresi nel suo piano di studi", si laurei nella facoltà una certa figura professionale (es. Ingegnere informatico specializzato in e-Learning).

In questo contesto a noi interessano soltanto gli obiettivi formativi di un corso, di un modulo e principalmente di un learning object.

Gli obiettivi formativi di un corso sono pochi ed in generale "abbastanza autoesplicativi"; ad esempio il corso "Gli attori del sistema formativo-comunicazione e apprendimento" potrebbe avere come obiettivo la comprensione e lo sviluppo di una capacità di gestione delle dinamiche interpersonali collegate ai ruoli, nuovi o differenti, coperti dai diversi attori del processo formativo; la sperimentazione dal vivo attraverso l'analisi delle

situazioni e dei comportamenti all'interno del corso stesso, di come le persone ed i gruppi reagiscono agli stimoli delle tecnologie e delle nuove situazioni, cogliendone le implicazioni didattiche e organizzative.

Il corso FIODL aveva come obiettivo principale il fornire ai discenti una preparazione, sia teorico sia pratica, sui fondamenti dell'informatica e sul linguaggio C, tale da poter superare l'esame finale virtuale (Prova Finale) con almeno il minimo dei voti diciotto (*Obiettivo corso misurabile*).

Analizziamo adesso gli obiettivi formativi di un Modulo, di un SottoModulo e dei Learning Objects.

I Learning Objects, di per se autoconsistenti, sono aggregati fra loro al fine di creare un SottoModulo che dev'essere completo in termini contenutistici, funzionali e formativi in funzione di specifici obiettivi didattici, finalizzati al raggiungimento dell'obiettivo didattico del Modulo.

I Moduli sono aggregati fra loro al fine del raggiungimento dell'obiettivo didattico generale del Corso.

Possiamo notare che per quanto riguarda gli obiettivi didattici si ha una gerarchia dettata anch'essa dal nostro schema a livelli della figura 1.4 (*obiettivi didattici gerarchici*).

2 L'esperimento eseguito: Corso FIODL

2.1 Ambiente di lavoro

Il corso sperimentale di didattica a distanza FIODL, la cui pagina di presentazione è mostrata nella figura 2.1 ed è reperibile all'indirizzo: <http://digilander.libero.it/capano/index.html>,

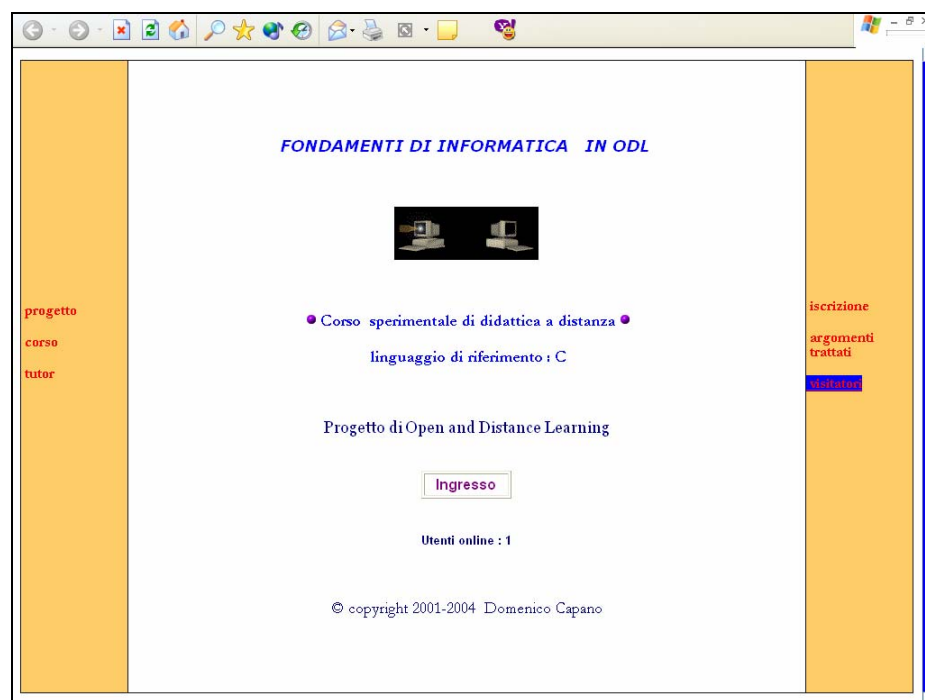


Fig. 2.1 - (Home page corso FIODL)

è iniziato il 15 luglio 2001 e terminato il 15 dicembre 2001.

Il corso è stato proposto a dodici studenti di Fondamenti di Informatica I e II del Consorzio NETTUNO, il cui docente era il relatore della tesi di laurea già menzionata.

Per gli studenti di Fondamenti I, FIODL si articolava in sedici lezioni e relativo corredo³⁸ "le prime sedici lezioni del corso" corso1,1; per gli studenti di Fondamenti II, FIODL si articolava in nove lezioni "dalla lezione n.13 alla lezione n. 21 e relativo corredo"³⁹, corso1,2, con un'ampia appendice, facoltativa, consistente nelle prime 12 lezioni del medesimo".

2.2 Obiettivi ed aspetti dell'esperimento

Uno dei nostri obiettivi è stato *sperimentare l'efficacia didattica del corso* di Fondamenti di Informatica in ODL, su un campione alquanto omogeneo di allievi, che doveva sostenere l'esame di fondamenti 1 e fondamenti 2 nell'ambito del Consorzio NET.T.UN.O.

Gli allievi in questione, per la strutturazione didattica, al tempo, del Consorzio suddetto, non avevano un corso *face to face* da seguire, disponevano soltanto di lezioni⁴⁰, su videocassetta, per cui FIODL ha rappresentato, per loro, una metodologia didattica interattiva ed innovativa rispetto a quelle disponibili allora.

Nel progetto si è insistito su argomenti che bene potessero integrare il corso tradizionale di NETTUNO (40 lezioni per corso in videocassette).

(Attualmente, Anna Maria Garito direttore del Consorzio Nettuno spiega, in un articolo⁴¹, «Siamo riusciti a sviluppare un modello didattico di grande impatto, il nostro portale è ricchissimo di contenuti, abbiamo 450 moduli didattici⁴² 20.000 ore di video lezioni trasmesse sia via satellite sui canali Rai Nettunosat 1 e 2, sia via Internet, e 20.000 ore di esercitazioni su Internet. Possiamo dire di essere stati l'esempio per creare l'università europea a distanza e l'università del Mediterraneo». Le affermazioni, del direttore Anna Maria Garito, sulle potenzialità del Consorzio via Internet, attualmente, ci sembrano troppo ottimistiche; certo è che fra le soluzioni didattiche

³⁸ Vedere la Tabella 1.4

³⁹ Vedere la Tabella 1.5

⁴⁰ Fondamenti di Informatica I e II, prof. A. R. Meo e M. Mezzalana del politecnico di Torino

⁴¹ Laura Kiss, *Cresce l'Italia nel mondo dell'e-Learning*, Affari&Finanza, 5 aprile 2004

⁴² Col termine moduli didattici si intende nell'articolo, quel che noi abbiamo chiamato Corsi.

disponibili in Italia in modalità *blended e-Learning*, Nettuno è un esempio buono, imitato in tutta Europa e che le iscrizioni, ai 24 corsi di laurea disponibili, sono cresciute notevolmente nel 2004).



Fig. 2.2 - (Home page sito web del Consorzio NETTUNO) ⁴³

Quindi integrazione degli argomenti dei corsi ufficiali e preparazione per l'esame ufficiale (face to face) di fondamenti 1 e 2 con il loro docente e mentor di questo esperimento di e-Learning.

Un altro aspetto del corso curato è stata la *semplicità dell'impostazione e fruizione delle pagine web*, in modo da non scoraggiare al primo impatto l'allievo avente poca familiarità con Internet.

Infatti è basilare in un corso a distanza, come nell'e-Learning nella sua più ampia accezione, individuare il target d'utilizzatori delle risorse che si rendono disponibili online.

Con nostra sorpresa, nonostante l'url del sito non fosse stato inserito nei motori di ricerca, sono arrivate tante iscrizioni da persone da tutta Italia, (avendoci pensato gli *spider* di alcuni motori di ricerca ad indicizzare le

⁴³ Consorzio nato nel 1970 , <http://www.uninettuno.it>

pagine del sito) che purtroppo per la scelta operata di sperimentazione sul campione suddetto siamo stati costretti a rifiutare.

2.3 Organizzazione

L'accesso ai contenuti salienti del corso era protetto da uno username e da una password, in virtù del tipo di scelta fatta per la fruizione del medesimo, come avviene tra l'altro oggi in tutte le piattaforme didattiche moderne.

Accedendo alla home page del sito si poteva avere visione dei contenuti soltanto delle pagine, home, progetto, tutors, visitatori, argomenti trattati e iscrizione.

Siccome “or non è più quel tempo e quell'età” il restante contenuto si può visualizzare attraverso lo user name:zxck e la password: mail_pass, accedendo da <http://digilander.libero.it/capano/index.html> e si entra nella pagina web frameset FIODL mostrata nella figura 2.3.

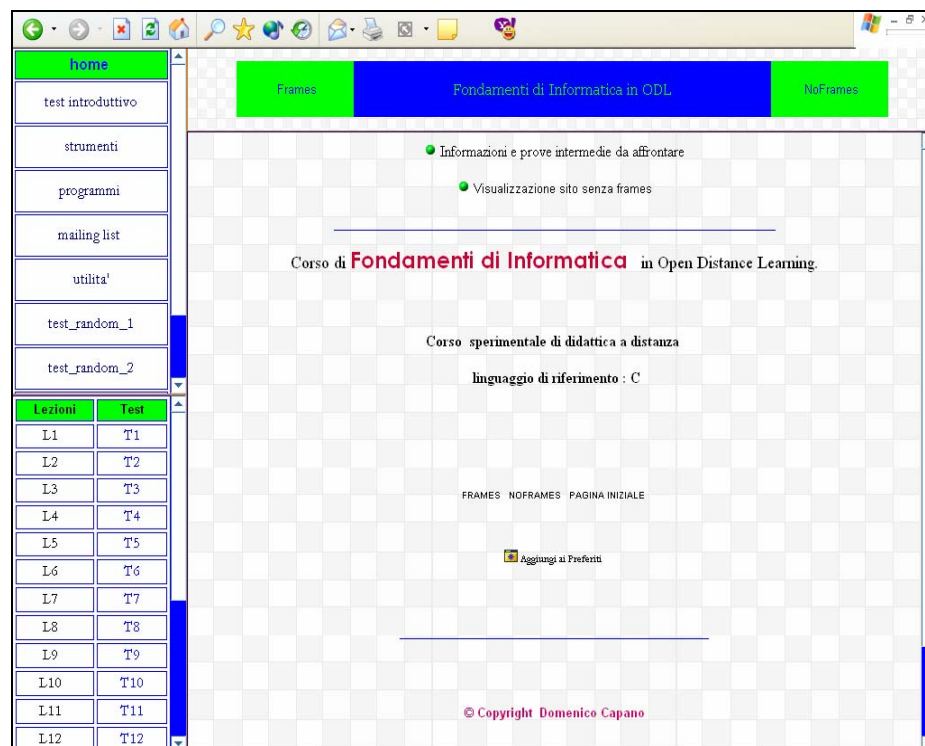
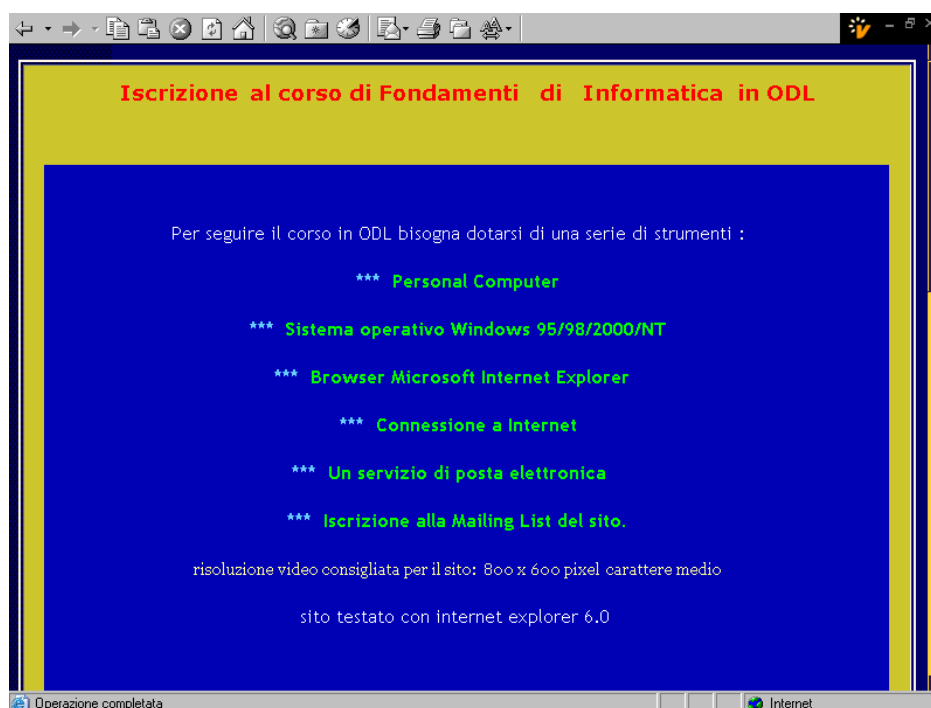


Fig. 2.3 - (Frameset FIODL)

2.3.1 Lista di iscrizione e conoscenza iniziale

Riportiamo nelle seguenti figure 2.4 parte del modulo di iscrizione che abbiamo predisposto nel sito (la nostra semplice piattaforma LCMS), per raccogliere le richieste di iscrizione al corso da parte degli utenti.

Nel caso in cui gli utenti iscritti appartenessero al target di allievi prefissato, ricevevano in risposta una e-mail con *user name* e *password*, e con l'invito a compilare ed inviare, come primo passo del corso a cui si erano iscritti, il corrispondente test introduttivo predisposto nel sito web.



The image shows a web browser window with a registration form titled "Modulo di iscrizione". The form is set against a dark blue background with yellow borders. It contains the following fields:

- Nome :
- Cognome :
- Città':
- Provincia:
- E-mail :
- Matricola :
- Corso :

Below the fields, there is a red rectangular area on the left and a yellow rectangular area on the right containing two buttons: "Invii" and "Cancelli". At the bottom of the form area, the text "pagina iniziale" is visible. The browser's status bar at the bottom shows "Operazione completata" on the left and "Internet" on the right.

Figure 2.4 - (Parte del modulo di iscrizione 2)

Nel modulo del sito visibile a tutta l'utenza web site, come si può parzialmente notare dalla figure 2.4 sopra mostrate, erano indicati gli strumenti cui era necessario dotarsi al fine di seguire correttamente il corso a distanza.

Gli altri strumenti cui dotarsi (es. compilatore per linguaggio C) sono stati indicati e resi disponibili online soltanto a coloro, che hanno avuto accettata l'iscrizione, appartenendo, come già detto, al target di discenti scelto in fase di Analysis.

2.3.2 Elenco degli iscritti ai corsi

Presentiamo in tabella 2.1 l'elenco degli iscritti, che hanno inviato il test introduttivo, in ordine di iscrizione al corso FIODL

Tab. 2.1 - Elenco iscritti corso FIODL

Ord. di Iscrizione	Sordine,corso	Cognome	Nome	Corso seguito
1	S1,1	Mon***	Federico	Fondamenti 1
2	S2,1	Giu***	Fabio	Fondamenti 1
3	S1,2	Nap***	Sabatino	Fondamenti 2
4	S3,1	Cer***	Andrea	Fondamenti 1
5	S2,2	Gri***	Agatino	Fondamenti 2
6	S4,1	Ber***	Costantino	Fondamenti 1
7	S3,2	Gre***	Ranieri	Fondamenti 2
8	S4,2	Pic***	Fabio	Fondamenti 2
9	S5,2	Bel***	Michele	Fondamenti 2
10	S5,1	D'E***	Fabio	Fondamenti 1
11	S6,1	Fai***	Vincenzo	Fondamenti 1
12	S6,2	Con***	Domenico	Fondamenti 2

Si_jc indica lo studente i-esimo del corso j_c-esimo con $c=1^{44}$ e $j=1,2$. L'indice i in questo caso è funzione di j_c ($i=f(j_c)$), poiché in generale ogni corso_jc del catalogo c potrebbe avere un diverso numero di studenti iscritti. Nel nostro caso avevamo un uguale numero di iscritti per i due corsi; per il corso₁,1 sei iscritti quindi $i=1 \dots 6$; per il corso₁,2 sei iscritti $i=1 \dots 6$.

⁴⁴ c è l'indice dei cataloghi contenenti corsi.

Precisiamo che, fra coloro che si sono iscritti al corso, due learners hanno compilato ed inviato i test introduttivi soltanto (hanno preso il materiale, a disposizione nel sito, che era protetto da password) ed un altro learner ha anch'esso incrementato il *drop out* (gli abbandoni) poiché ha trovato la propria strada, adesso è in seminario.

2.3.3 Test Introduttivi

Per valutare il livello di preparazione di partenza degli allievi abbiamo utilizzato due Test introduttivi⁴⁵ (Ricordiamo che un Test Introduttivo lo abbiamo definito, nel capitolo 1, come un *particolare Modulo* “il primo modulo erogato” *costituito da un unico SottoModulo*); un Test riservato agli studenti iscritti a Fondamenti I composto da 20 domande e l'altro riservato agli studenti di Fondamenti II composto da 30 domande. I due tipi di test naturalmente avevano difficoltà diverse.

Nelle fig. 2.5, e 2.6 riportiamo parte del test di Fondamenti I.

test iniziale I - Microsoft Internet Explorer fornito da Virgilio

FRAMES NOFRAMES

Test introduttivo al corso di FI-ODL

"per gli studenti di fondamenti di informatica I"

Questo test introduttivo serve per valutare il livello base della preparazione dei partecipanti al corso.

Non ci si aspetta che tutti rispondano esattamente a tutte le domande ...

alla fine del corso un altro test verificherà se la preparazione ha avuto i miglioramenti attesi, cioè se il corso è stato utile.

I dati risultanti dal test non verranno divulgati, se non in forma aggregata.

Vi invitiamo alla compilazione e Buon lavoro.

Compilate il questionario in tutte le sue parti ed inviate subito.

Cognome :

⁴⁵ Utilizzeremo nel Capitolo 4 i risultati di questi test introduttivi per fare delle analisi comparative. Nel quinto capitolo un test introduttivo *unico* può essere utilizzato per dirottare il learner verso un percorso di apprendimento personalizzato di sua comprensibilità.

Fig. 2.5 - (Test Introduttivo F1 (a))

test iniziale II - Microsoft Internet Explorer fornito da Virgilio

Cognome :

Nome :

Num. di matricola :

E-mail :

Corso :

Leggete attentamente le domande del test e date la/le vostra/e risposta/e .

1) Per quanto tempo la memoria RAM di un personal computer conserva le informazioni?

- ☐ Finchè lo hard disk funziona in modo corretto ossia non vi sono settori danneggiati
- ☐ Fintanto che il computer è alimentato
- ☐ Fintanto che la scheda madre del computer è alimentata con una batteria tampone

2) Che cosa e' la ALU ?

- ☐ Il processore del computer

test iniziale I - Microsoft Internet Explorer fornito da Virgilio

19) Perché nel passaggio di un array ad una funzione, quest'ultimo viene sempre passato per indirizzo nel linguaggio C?

20) Si assuma di rappresentare numeri in complemento a due. Se la rappresentazione di un numero mediante quattro bit e' 1101, la sua rappresentazione mediante sei bit e':

- ☐ 001101
- ☐ 011101
- ☐ 101101
- ☐ 111101

Fondamenti di Informatica in ODL

Start | ZonaAlarm | accesso riservat... | bozzalesi1 - Micr... | test iniziale I ... | 4.17

Figure 2.6 - (Test Introduttivo F1 (b, c))

Riportiamo di seguito un esempio di *risposta* data da un learner, al SottoModulo Test introduttivo $\equiv$ (SMod1,1,1,1) per Fondamenti 1, pervenutaci nelle *e-mail-tutors*.

Cognome = Cer*** Nome=And***

Numero di matricola = 09112***

E-mail = and***.cer***@tiscalinet.it

Corso = Fondamenti di Informatica 1

1) Per quanto tempo la memoria RAM di un personal computer conserva le informazioni? = Fintanto che il computer è alimentato **2)** Che cosa è la ALU? = L' unità aritmetica e logica **3)** Il seguente numerale A1F0 in base 16 a quale numero in base 10 corrisponde? = 41456 **4)** Il seguente pseudocodice supponendo $n > 0$: quale problema risolve ? = non funziona in quanto nella dichiarazione alla variabile somma abbiamo assegnato il valore zero. **5)** La rappresentazione degli interi relativi X , nell'intervallo $[-(b^n/2), (b^n/2)$), tramite il residuo modulo b^n di X , è detta: = Rappresentazione in complemento alla base b **6) a)** Quali linguaggi di programmazione conosce? = Nessuno **6) b)** Inserite eventuali altri linguaggi di vostra conoscenza = Ho frequentato un corso di Basic più di dieci anni fa **7)** Quale problema risolve il seguente algoritmo, se r è un numero intero maggiore di 1:

1) leggi r ; 2) $\text{cont} \leftarrow 1$; 3) $\text{cont} = r * \text{cont}$; 4) $\text{somma} = 1$; 5) se ($\text{cont} \leq r$)
{ $\text{somma} \leftarrow \text{somma} + \text{cont}$; $\text{cont} \leftarrow \text{cont} * 2$; } 6) stampa somma ;

= Stampa, in somma , il valore di $2r$ **8) a)** Quali fra i sottoelencati sono dei sistemi operativi e che conosce = UNIX **8) b)** Quali fra i sottoelencati sono dei sistemi operativi e che conosce = DOS **8) c)** Quali fra i sottoelencati sono dei sistemi operativi e che conosce = WINDOWS 98 **9)** La seguente funzione in C: $\text{int F(int k) \{if(k=0) return 1;else return k * F(k-1);\}}$ che tipo di ricorsione usa e per calcolare che cosa? = usa la ricorsione per calcolare il fattoriale di k , ma la funzione è non corretta. **10)** Come può essere usata una stringa in un programma in linguaggio C= xxx **11) a)** Scrivete l'intestazione di una funzione chiamata *scambia* che riceva come parametri due puntatori ai numeri in virgola mobile n e m e che non restituisca alcun valore = **11) b)** Scrivete il prototipo per la funzione della parte a) = xxx **12)a)** Un array monodimensionale può immagazzinare valori di molti tipi differenti = scelga; motivazione 12 a) = xxx **12) b)** La funzione, che usa l'algoritmo di ordinamento selection-sort, per ordinare un array di 5 elementi interi, utilizza nel caso peggiore 10 confronti.= scelga; motivazione 12 b) = xxx **12)c)**

Supponiamo di avere la seguente dichiarazione `int new_array[15]`; è corretto leggere da input gli elementi del vettore attraverso la seguente istruzione:

`for(i=0;i<15;i++) scanf("%i",&new_array[i]);` con `i` definito di tipo intero. = scelga motivazione 12 c) = xxx. **12)d)** Avendo la seguente dichiarazione: `int *new_array`; è corretto leggere da input gli elementi del vettore attraverso la seguente istruzione: `for(i=0;i<15;i++) scanf("%d", (new_array+i));` con `i` definito di tipo intero. = scelga motivazione 12 d) = xxx. **13)** Cosa fa il programma in linguaggio C seguente ... = xxx **14)** Un vettore elementare di nome `vector`, di 10 caratteri, come si può dichiarare nel linguaggio C? = xxx

15) la seguente dichiarazione `int k[8]`; cosa rappresenta? = un numero intero di 8 cifre **16)** (11001.1001) in base 2 corrisponde a ? in base 10 = 11001.1001

17) B1E7 esadecimale corrisponde a ? in base 2 = 1011000111100111

18) Cosa succede ai parametri attuali, di una funzione, dopo la cessata attivazione: nel passaggio per valore ed in quello per variabile (od indirizzo)? = xxx. **19)** Perché nel passaggio di un array ad una funzione, quest'ultimo è sempre passato per indirizzo nel linguaggio C? = xxx **20)** Si assuma di rappresentare numeri in complemento a due. Se la rappresentazione di un numero mediante quattro bit è 1101, la sua rappresentazione mediante sei bit è: = 001101

Tempo di inizio : 4:38:25 P.M. Tempo finale : 5:19:50 P.M. **Tempo per eseguire il Test : 00:41:25**

Il learner Cer*** ha impiegato 41' e 25'' per rispondere alle domande del test introduttivo e nonostante il tempo dedicato fosse sufficiente alla completa risoluzione del test ha ottenuto⁴⁶ un punteggio pari a: **5/30**.

Da tale punteggio si deduce che egli aveva una preparazione iniziale bassa sugli argomenti trattati dal corso F1, come d'altronde si intuisce da alcune risposte da lui date.

Nel capitolo 4 mostreremo, fra gli altri risultati prodotti dal corso, i grafici con i voti riportati dai learners nei Test Introduttivi ed in quelli Finali al fine di simulare il profitto didattico raggiunto dai learners.

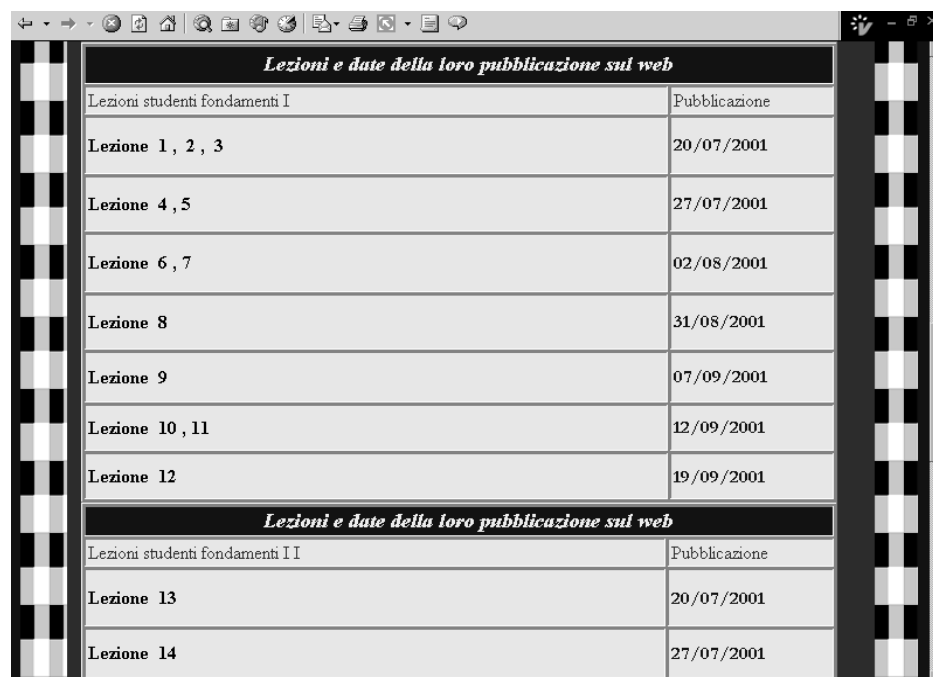
⁴⁶ La correzione dei test è stata fatta dal *tutor* manualmente; nelle odierne *piattaforme* e-Learning è possibile prevedere le correzioni automatiche dei test introduttivi. Notare che il tutor è la figura di ausilio per coadiuvare il processo di apprendimento dei learners fornendo stimoli di tipo collaborativo, che si affianca al docente del corso ed al *mentor*, ma non li sostituisce mai.

3 Svolgimento del Corso

3.1 Lezioni

Dopo l'invio del Test introduttivo da parte di tutti i learners abbiamo pubblicato nel web le lezioni, che abbiamo detto essere nel nostro schema a livelli dei *SottoModuli*.

Nelle Fig.re 3.1 mostriamo le lezioni divise per corso (studenti fondamentali I e fondamentali II) e le date della loro pubblicazione sul web.



The image is a screenshot of a web browser window displaying a table titled "Lezioni e date della loro pubblicazione sul web". The table is divided into two sections, one for "Lezioni studenti fondamentali I" and one for "Lezioni studenti fondamentali II". Each section has two columns: "Lezione" and "Pubblicazione".

<i>Lezioni e date della loro pubblicazione sul web</i>	
Lezioni studenti fondamentali I	Pubblicazione
Lezione 1 , 2 , 3	20/07/2001
Lezione 4 , 5	27/07/2001
Lezione 6 , 7	02/08/2001
Lezione 8	31/08/2001
Lezione 9	07/09/2001
Lezione 10 , 11	12/09/2001
Lezione 12	19/09/2001
<i>Lezioni e date della loro pubblicazione sul web</i>	
Lezioni studenti fondamentali II	Pubblicazione
Lezione 13	20/07/2001
Lezione 14	27/07/2001

Lezione 15 , 16	02/08/2001
Lezione 17	31/08/2001
Lezione 18	06/09/2001
Lezione 19	13/09/2001
Lezione 20	20/09/2001
Lezione 21	29/09/2001

FRAMES NOFRAMES

Eventuali comunicazioni di interesse collettivo verranno visualizzate in questo spazio

* Il corso di fondamentali di informatica in ODL e' iniziato il 18 luglio 2001, fino al 2/08/2001: prima parte.

* La seconda parte del corso e' ripresa il 31/08/2001.

* Al termine del corso, si prega i partecipanti di compilare un questionario finale di valutazione del medesimo.

* Svolgete le prove intermedie con una maggiore tempestività'.

Figure 3.1 - (Lezioni e date della loro pubblicazione sul web)

3.2 Comunicazione

Il mezzo principale con cui si è svolta la comunicazione (fra studenti e tutors, fra studente e studenti ..) è stato attraverso la **mailing list**, cioè una raccolta d'indirizzi di posta elettronica di tutti i protagonisti del corso.

Nel sito abbiamo dato la possibilità di inviare messaggi, ad un *singolo iscritto* alla mailing list, mediante selezione da un menu a tendina del nominativo dell'iscritto (Fig. 3.3), ad un *gruppetto* da selezionare a piacimento mediante checkbox (Fig. 3.4) ed a *tutti gli iscritti* alla mailing list (Fig. 3.5).

In quest'ultimo caso nel sito vi era predisposto uno script in linguaggio Javascript che permetteva di superare il limite, al max numero di indirizzi inviabili con un solo click, attraverso un trucchetto di invio a più riprese del medesimo messaggio (indirizzi divisi in gruppi di dimensione pari al max numero inviabile). Tanti click per quanti sono i gruppi⁴⁷.

⁴⁷ Per visionare il codice andare nella sezione 9.1.2

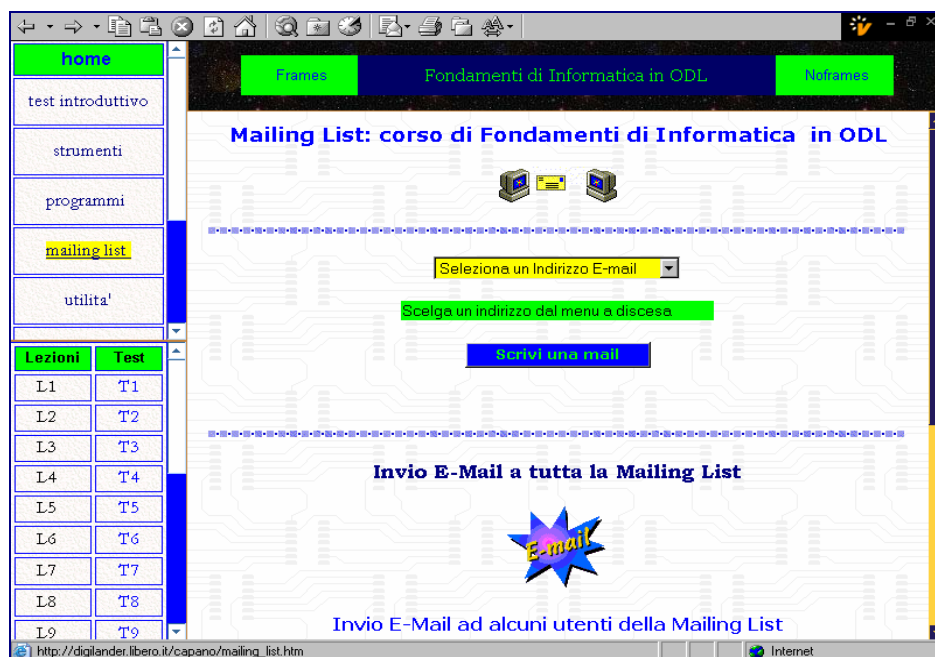


Fig. 3.2 - (Invio E-mail ad utente selezionato dal menu a tendina)

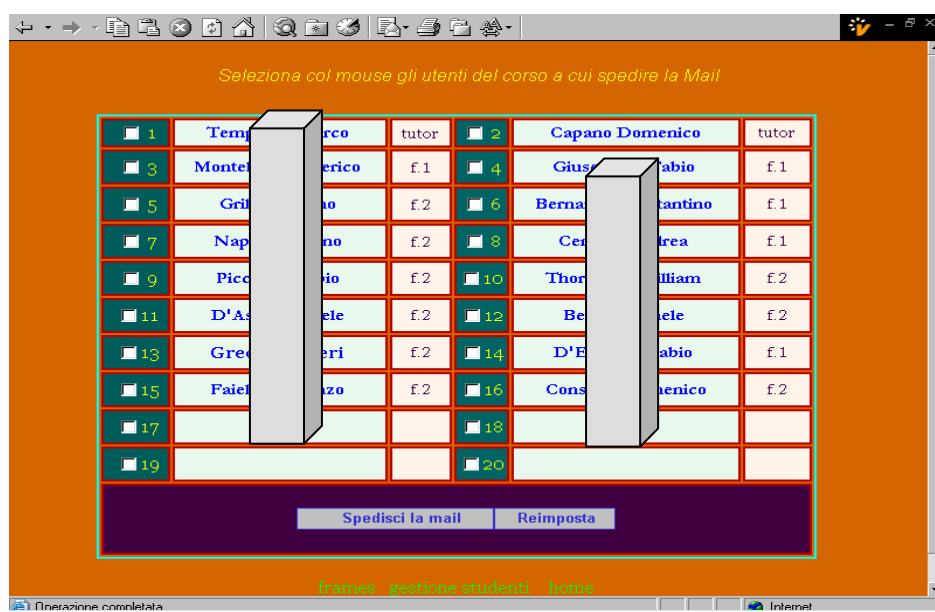


Fig. 3.3 - (Invio E-Mail ad alcuni utenti scelti)

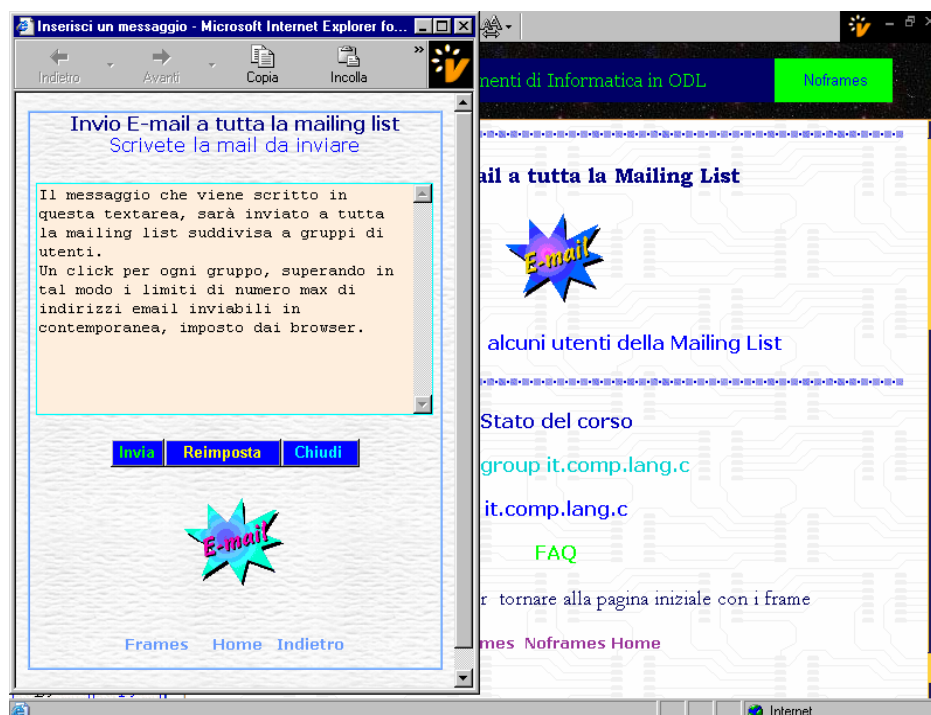


Fig. 3.4 - (Invio E-Mail a tutta la Mailing list)

Un altro strumento disponibile nel sito è la **chat** non vocale fornita da Bravenet mostrata in figura. 3.5.

Con tale mezzo di *comunicazione sincrona* si è creato, con gli studenti che l'hanno utilizzata, un rapporto interattivo in tempo reale, da loro molto apprezzato.

La chat nell'e-Learning è uno dei principali strumenti di interazione, utilizzati nei modelli didattici contenenti il *cooperative learning*, attraverso cui i discenti cooperano attivamente alla crescita didattica dell'aula virtuale. L'apprendimento cooperativo e collaborativo sono parte fondamentale nel modello di apprendimento *learner centered*, utilizzato nei corsi di formazione in modalità e-Learning, in linea con la teoria del *costuttivismo*, contrapposto al vecchio modello *teacher centered*, tipico dei corsi frontali od in presenza.



Fig. 3.5 - (Chat non vocale)

Nel corso FIODL (con tutti i limiti della piattaforma web (LMS) utilizzata) abbiamo realizzato un *modello di comunicazione uno a uno ed uno a molti* tra tutor, discenti e docente, valido in generale nei progetti di un corso di formazione in modalità e-Learning, così come evidenziato nella figura 3.6.

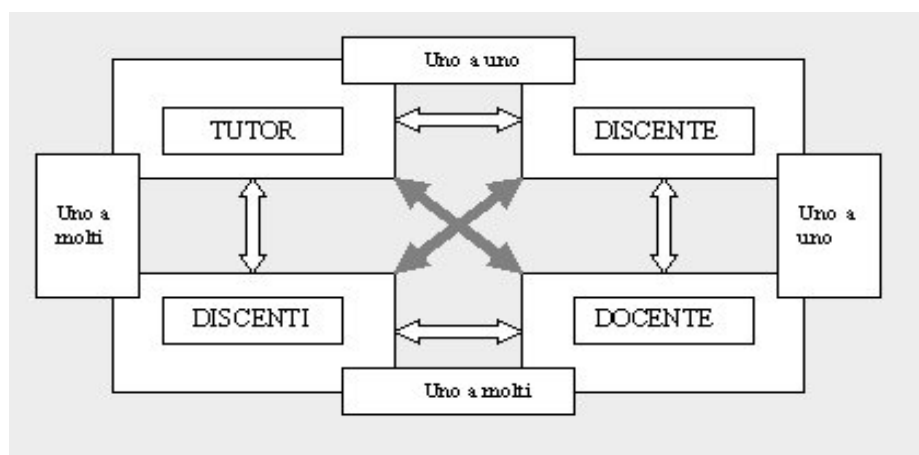
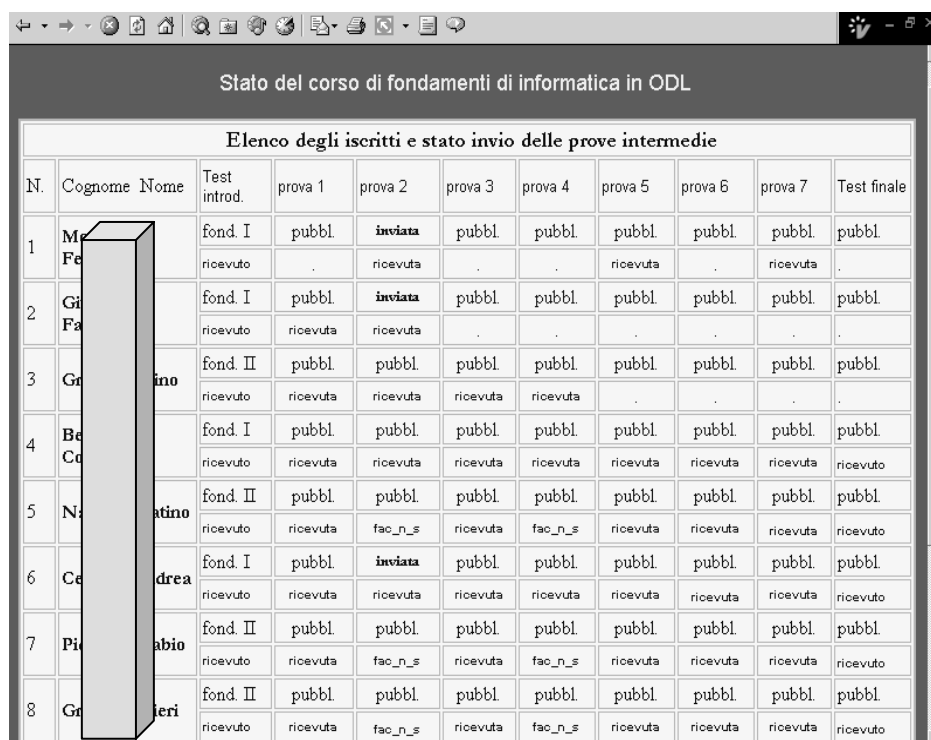


Fig. 3.6 - (Modello di comunicazione)

3.2.1 Stato del Corso

Gli studenti potevano prendere visione dell'andamento dei lavori nella pagina raffigurante lo stato del corso, che era aggiornata di volta in volta e che fungeva da *bacheca elettronica (comunicazione asincrona)*.



N.	Cognome Nome	Test introd.	prova 1	prova 2	prova 3	prova 4	prova 5	prova 6	prova 7	Test finale
1	M. Fe	fond. I	pubbl.	inviata	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	.	ricevuta	.	.	ricevuta	.	ricevuta	.
2	Gi. Fa	fond. I	pubbl.	inviata	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	ricevuta	ricevuta	.	.	.	.	.	.
3	Gr. tino	fond. II	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	ricevuta	ricevuta	ricevuta	ricevuta	.	.	.	.
4	Be. Co	fond. I	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	ricevuta	ricevuta	ricevuta	ricevuta	ricevuta	ricevuta	ricevuta	ricevuto
5	N. atino	fond. II	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	ricevuta	fac_n_s	ricevuta	fac_n_s	ricevuta	ricevuta	ricevuta	ricevuto
6	Ce. drea	fond. I	pubbl.	inviata	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	ricevuta	ricevuta	ricevuta	ricevuta	ricevuta	ricevuta	ricevuta	ricevuto
7	Pic. abio	fond. II	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	ricevuta	fac_n_s	ricevuta	fac_n_s	ricevuta	ricevuta	ricevuta	ricevuto
8	Gr. eri	fond. II	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.	pubbl.
		ricevuto	ricevuta	fac_n_s	ricevuta	fac_n_s	ricevuta	ricevuta	ricevuta	ricevuto

Fig. 3.7 - (Stato del corso)

Nel corso a distanza attraverso i mezzi di comunicazione sincrona (chats) ed asincrona (bacheche, mailing list,..) abbiamo creato un'*aula virtuale*⁴⁸ molto importante per far interagire allievi con allievi ed allievi con i tutors al fine di attenuare il senso di isolamento dei learners tipico un un corso a distanza ed applicare il *learning collaborativo* che accresce le competenze di ogni partecipante migliorando al contempo la qualità didattica del corso oltre che fungere da stimolo didattico. Abbiamo inoltre inserito nel sito una sezione contenente le **FAQ** (Frequently Asked Questions) cresciute in itinere.

⁴⁸ In generale, nella terminologia dell'e-Learning con aula virtuale s'intende l'ambiente virtuale creato dagli utenti in comunicazione sincrona. Qui intendiamo l'ambiente virtuale creato da tutti i protagonisti del corso online ossia una porzione consistente della piattaforma e-learning.

3.2.2 Autovalutazione dei partecipanti al corso

L'autovalutazione degli studenti è avvenuta attraverso i test di autovalutazione.

Abbiamo usato un test per ogni lezione, del tipo a risposta multipla, ad opzioni, a completamento e misto.

Abbiamo dato la possibilità di avere anche test random selezionati fra i suddetti e due test di autovalutazione finali con domande scelte randomicamente sull'intero corso.

Per maggiori dettagli sui test vedi il paragrafo 3.3.

In generale nei corsi di formazione in modalità e-Learning abbiamo come tipi di test di autovalutazione, quelli con le:

- *domande con risposte multiple* di cui una o più di una può essere la risposta esatta;
- domande con risposte del tipo vero-falso;
- *domande con risposta aperta*, ovvero una casella di testo nella quale è possibile scrivere la risposta.
- *Testo con parole omesse*, ovvero un testo dove alcune parole sono sostituite da uno spazio vuoto; in questo caso l'esaminando dovrà ripristinare le parole omesse negli spazi vuoti.
- *Associazione di parole*, ovvero coppie di parole di cui bisogna indovinare la giusta corrispondenza.⁴⁹

3.2.3 Valutazione degli studenti

La valutazione degli studenti è stata da noi attuata attraverso le *Prove Intermedie* ed il *Test Finale*.

Le prove intermedie presentate consistevano in dei *problemi di programmazione in linguaggio C*, generalmente, da svolgere ed inviare attraverso posta elettronica nelle caselle e-mails del tutor e del mentor del corso.

Nell'appendice B, riportiamo integralmente uno dei due di Test Finali di valutazione del livello di apprendimento dei discenti a loro proposti nel corso FIODL.

⁴⁹ Un generatore automatico di test di autovalutazione è quello realizzato da Luca Galli *QuizFaber*, una versione, anche free, è reperibile all'indirizzo: www.lucagalli.net

3.2.4 Valutazione del corso

Alla fine del corso gli studenti hanno valutato il livello qualitativo del corso FIODL seguito compilando il questionario di valutazione del corso composto da 24 domande, riportato integralmente in appendice C.

Di seguito mostriamo la prima videata del questionario di valutazione del corso FIODL visibile nel web.

Valutazione del corso FI-ODL

[Frames](#) [Home](#)

In questo questionario le vostre risposte, servono a valutare il corso ed i suoi docenti.
Completarlo in tutte le sue parti.
Le vostre risposte ci saranno molto utili per capire se abbiamo fatto errori e come eventualmente correggerli in futuro.
Osservazioni e commenti, anche non direttamente suscitati dalle domande, saranno graditi.

1) Perché hai deciso di seguire il corso FI-ODL ?
(selezionare tutte le opzioni di interesse)

☐ Mi interessava sperimentare la didattica a distanza

☐ Mi interessavano gli argomenti trattati nel corso

☐ Mi interessava avere materiale aggiuntivo sulla materia

Altri motivi

2) La tua motivazione prima di iniziare il corso era:

☐ Molto alta

☐ Alta

☐ Sufficiente

☐ Scarsa

Fig. 3.8 - (Videata Questionario di valutazione Corso FIODL)

3.3 Test di autovalutazione

Nel corso FIODL abbiamo predisposto un test di autovalutazione per ogni lezione, da farsi al termine dello studio della lezione.

Riportiamo in figura. 3.9 un esempio di test di autovalutazione del tipo a completamento, in figura 3.10 un test di autovalutazione ad opzioni ed in figura 3.11 un test di autovalutazione del tipo a risposta multipla.

Ogni test dava all'allievo le informazioni sul risultato conseguito e sul tempo impiegato per lo svolgimento attraverso delle finestre di alerts.

The screenshot shows a web browser window with the title 'Fondamenti di Informatica in ODL'. The page is titled 'Test a completamento 14' and contains the instruction 'Completate ognuna delle seguenti domande:'. There are six questions, each with a text input field for the answer. The questions are:

- 1) L'operatore restituisce la locazione di memoria in cui e' immagazzinato il suo operando.
- 2) I tre valori che possono essere utilizzati per inizializzare un puntatore sono , o .
- 3) La funzione scanf necessita degli poiche' essa deve modificare gli argomenti e quindi bisogna simulare per essi il passaggio per indirizzo.
- 4) Con la scrittura `punt=&cont;` abbiamo una istruzione che pone nella variabile punt l' della variabile cont.
- 5) Avendo `int x,y; int *indir;` l'assegnazione assegna al puntatore indir l'indirizzo di x.
- 6) Nelle funzioni il passaggio per indirizzo viene utilizzato per modificare il del .

At the bottom of the page, there are two buttons: 'Verifica' and 'Reimposta'.

Fig. 3.9 - (Test a completamento)

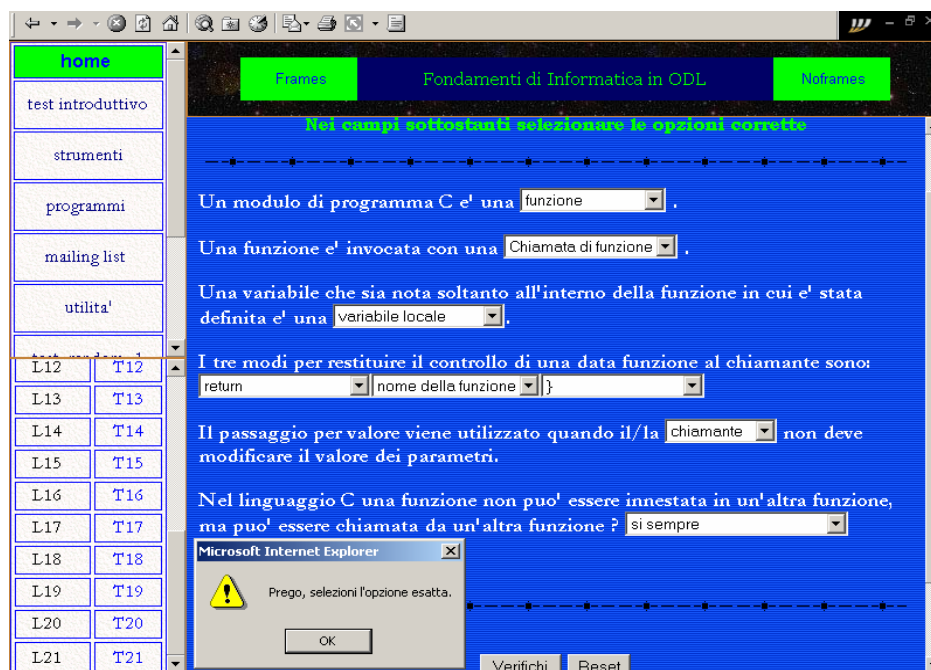


Fig. 3.10 - (Test ad opzioni)

Nella figura 3.11 mostriamo un test di autovalutazione a risposta multipla (una risposta soltanto è corretta).

Abbiamo utilizzato in tale test un numero di domande pari ad otto con 4 scelte possibili per domanda.

Nella figura si può vedere l'esito del test ottenuto cliccando sul button **Risultati**.

E' conteggiato in questi tipi di test il numero di domande affrontate dal learner ed il numero di risposte esatte ottenute; una finestra di alert dà la valutazione corrispondente, anche numerica in trentesimi se si risponde esattamente ad un congruo numero di domande.

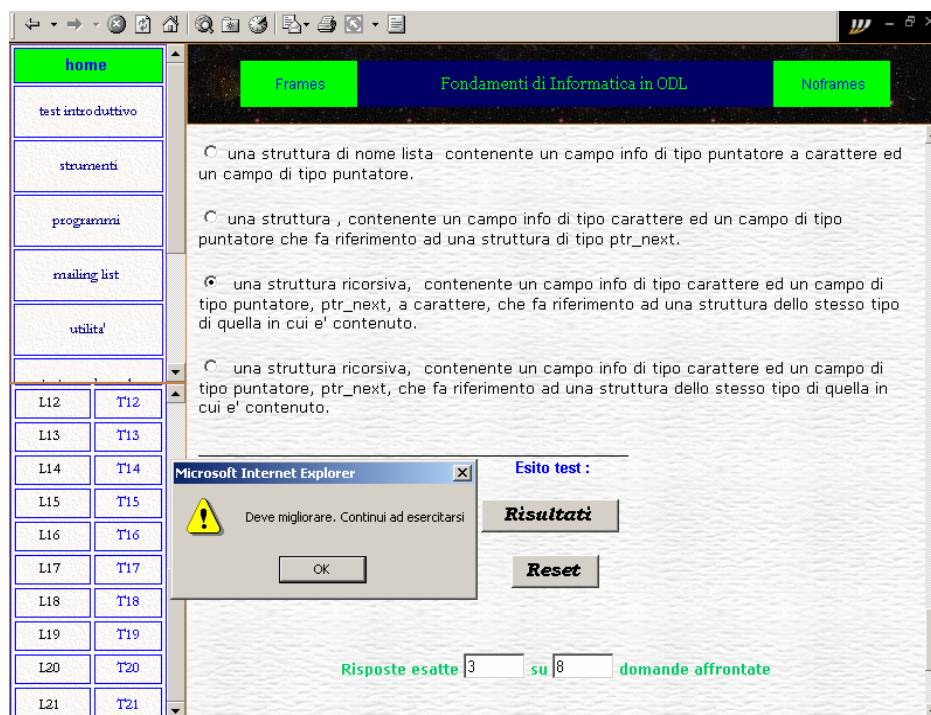


Fig. 3.11 - (Test a risposta multipla)

3.4 Prove intermedie assegnate

Corso sperimentale di didattica a distanza sul linguaggio C			
Data pubblicazione lezioni e prove per entrambi i corsi			
Prove intermedie da affrontare durante il corso FI-ODL per studenti fondamentali I			
Prove del corso	Vincolo	Pubblicazione	Invio al tutor
Test introduttivo I	Obbligatorio	15/07/2001	subito
Prova 1	Obbligatoria	29/07/2001	"
Prova 2	Obbligatoria	03/08/2001	"
Prova 3	Obbligatoria	04/09/2001	"
Prova 4	Facoltativa	20/09/2001	"
Prova 5	Obbligatoria	27/09/2001	"
Prova 6	Obbligatoria	05/10/2001	"
Prova 7	Obbligatoria	15/10/2001	"
Test finale I	Obbligatorio	08/11/2001	"
Prove intermedie da affrontare durante il corso FI-ODL per studenti fondamentali II			
Prove del corso	Vincolo	Pubblicazione	Invio al tutor
Test introduttivo II	Obbligatorio	15/07/2001	subito
Prova 1	Obbligatoria	23/07/2001	"
Prova 2	Facoltativa	29/07/2001	"
Prova 3	Obbligatoria	04/09/2001	"
Prova 4	Facoltativa	06/09/2001	"
Prova 5	Obbligatoria	23/09/2001	"
Prova 6	Obbligatoria	05/10/2001	"
Prova 7	Obbligatoria	15/10/2001	"
Test finale II	Obbligatorio	09/11/2001	"
Lezioni e date della loro pubblicazione sul web			
Lezioni studenti fondamentali I		Pubblicazione	
		08/07/2001	

Figure 3.12 - (Elenco Prove del corso FIODL)

Le Prove intermedie rappresentano l'interazione studente-(tutor e/o docente), *comunicazione uno a uno*. Una prova intermedia, per noi, è un esercizio, da svolgere, in linguaggio C. *Traccia dell'esercizio e sua soluzione da noi proposta* rappresentano un SottoModulo particolare costituito da un unico learning object.

La soluzione trovata dagli studenti era inserita nella textarea, predisposta nel modulo contenente la traccia della Prova, ed inviata tramite posta elettronica.

Esempio di soluzione del learner S3,1 Cer* alla Prova Intermedia 7**

Cognome=Ceras***

Nome=assalti frontali

Email=and***.ceras***@tiscalinet.it

Corso=f.informatica 1

Progettate un programma in linguaggio

C, il quale concateni due stringhe lette da input, di al più 35 caratteri ciascuna; e stampi le tre stringhe: prima stringa, seconda stringa, stringa concatenata.=

```
#include <stdio.h> /*soluzione con i vettori*/
#include <string.h>
#define N 36
void main () {
    char strin1[N],strin2[N];
    int n1,n2,i,j,z;
    printf("\n inserisci 2 stringhe:");
    scanf("\n%s%s",strin1,strin2);
    printf("\nlunghezza stringa1: %d",n1=strlen(strin1));
    printf("\nlunghezza stringa2: %d\n",n2=strlen(strin2));
    while ((n1>35)||(n2>35)){
        printf("\nuna o due stringhe troppo lunghe");
        printf("\n reinserisci 2 stringhe:");
        scanf("\n%s%s",strin1,strin2);
        printf("\nlunghezza stringa1: %d",n1=strlen(strin1));
        printf("\nlunghezza stringa2: %d\n",n2=strlen(strin2));
    }
    printf("\n%s\t%s\n",strin1,strin2);
    for (i=0,j=0; (i<n1)&&(j<n2); i++,j++)
        printf("%c%c",strin1[i],strin2[j]);
    if (n1>n2){
        while(n1>n2){
            printf("%c",strin1[i]);
            n2++;i++;
        }
    }
    if (n1<n2){
        while(n1<n2){
            printf("%c",strin2[j]);
            n1++;j++;
        }
    }
}
```



```

}
}

```

/*soluzione con i puntatori*/

```

#include <stdio.h>
#include <string.h>
#define N 36
void main () {
    char strin1[N],strin2[N],strinconc[N+N],*strinp1,*strinp2;
    int n1,n2,i,j=0,z=0;
    printf("\n inserisci 2 stringhe:");
    scanf("\n%s%s",strin1,strin2);
    printf("\nlunghezza stringa1: %d",n1=strlen(strin1));
    printf("\nlunghezza stringa2: %d\n",n2=strlen(strin2));
    while ((n1>35)||(n2>35)){
        printf("\nuna o due stringhe troppo lunghe");
        printf("\n reinserisci 2 stringhe:");
        scanf("\n%s%s",strin1,strin2);
        printf("\nlunghezza stringa1: %d",n1=strlen(strin1));
        printf("\nlunghezza stringa2: %d\n",n2=strlen(strin2));}
    printf("\nstringa1:%s\tstringa2:%s\n",strin1,strin2);
    strinp1=strin1;strinp2=strin2;
    if (n1>=n2){i=1;
    while(*strinp2!=0){
        strinconc[i]=*strinp2; i+=2; strinp2+=1;z++;
    }
    i=0;
    while(j<z){
        strinconc[i]=*strinp1; i+=2; strinp1+=1;j++;
    }
    while(*strinp1!=0){
        strinconc[i]=*strinp1; i+=1; strinp1+=1;
    }
    }
    else {i=0;
    while(*strinp1!=0){
        strinconc[i]=*strinp1; i+=2; strinp1+=1;j++;}
    i=1;
    while(j>z){
        strinconc[i]=*strinp2; i+=2; strinp2+=1;z++;
    }
    i--;
    while(*strinp2){
        strinconc[i]=*strinp2;strinp2+=1;i++;
    }
}

```

```

}
printf("\nla stringa concatenata è:\t");
for (i=0;i<(n1+n2); i++)
printf("%c",strinconc[i]);
}

```

- **Soluzione della Prova Intermedia 7 per F1 proposta nel corso1,1**

/*Progettate un programma in linguaggio C, il quale concateni due stringhe lette da input, di al più 35 caratteri ciascuna e stampi le tre stringhe: prima stringa, seconda stringa e terza stringa.*/

```

#include<stdio.h>
#define L_MAX 35

int main()
{
char car;
char prima_stringa[L_MAX+1], seconda_stringa[L_MAX+1];
char terza_stringa[2*L_MAX+1];
int i=0,j=0;
printf("\nScrivi la prima stringa ");
gets(prima_stringa);
printf("\nScrivi la seconda stringa ");
gets(seconda_stringa);
printf("\nho letto: %s e %s",prima_stringa,seconda_stringa);
printf("\nla stringa concatenata è\n");
for(;prima_stringa[i];i++)
terza_stringa[i]=prima_stringa[i];
for(;seconda_stringa[j];j++)
terza_stringa[i+j]=seconda_stringa[j]; /* copia seconda in terza a partire da
dove ero arrivato con l'indice i */
terza_stringa[i+j]='\0'; /* chiusura esplicita della terza stringa */
printf("\n%s\n",terza_stringa);
printf("\ninserite un carattere per terminare\n");
scanf("%c",&car);
return(0);
}

```

3.4.1 Cartella personale dello studente S3,1 protetta da password

The screenshot shows a web browser window with the address bar displaying `http://digilander.libero.it/capano/paginariservata/cilenpag.html`. The page content includes a greeting "Ciao Andrea!", a table of personal information, and a rating. A second browser window is open in the foreground, showing the solution for "Prova 7 per fondamenti 1".

Ciao Andrea!

Informazioni personali, non divulgabili agli altri studenti del corso		
Ce	voto	soluzione
Andrea	5/30	alla fine del corso
Test introduttivo	5/30	alla fine del corso
prova intermedia 1 obblig.	29/30	soluzione prova 1 fond. 1
prova intermedia 2 obblig.	30/30	soluzione prova 2 fond. 1
prova intermedia 3 obblig.	29/30	soluzione prova 3 fond. 1
prova intermedia 4 facolt.	29/30	soluzione prova 4 fond. 1
prova intermedia 5 obblig.	29/30	soluzione prova 5 fond. 1
prova intermedia 6 obblig.	30/30	soluzione prova 6 fond. 1
prova intermedia 7 obblig.	29/30	soluzione prova 7 fond. 1

Tuo Rating generale: + che ottimo !

Voto Test finale: 19/30

Soluzione Prova 7 per fondamenti 1

```
/*Progettate un programma in linguaggio C, il quale concateni due stringhe
lette da input, di al piu' 35 caratteri ciascuna e stampi le tre stringhe:
prima stringa, seconda stringa e terza stringa.*/

#include<stdio.h>
#define L_MAX 35
```

Fig. 3.13 - (Cartella personale del learner S3,1)

In generale, nei modelli di e-Learning centrati sull'utente, ogni discente ha la sua cartella/pagina personale e personalizzata che è indipendente dall'insegnamento o dalla materia.

3.4.2 Status di valutazione personale

Con la cartella personale, protetta da password lo studente ad ogni passo del corso che egli compiva, prove intermedie e test finale, era informato della valutazione data dal tutor; comparivano infatti nella cartella il suo rating attuale ed i consigli su cosa eventualmente insistere nello studio, oltre che la soluzione da noi proposta se aveva partecipato alla corrispondente prova. Naturalmente s'inviavano le correzioni delle prove tramite email, e li s'invitava a vedere il voto in esse riportato nella cartella personale.

Docente e tutor del corso, avevamo anche una nostra cartella personale; vedere la sezione 3.6 per avere visione del suo contenuto.

3.5 Test Random

Nel corso inoltre si è provveduto ad inserire due Test con domande random di ricapitolazione, di tutto il programma, per F1 e per F2 rispettivamente di 24 e 20 domande variabili sul range delle domande di tutto il Corso. Nelle figure 3.15 e 3.16 mostriamo una videata delle prime due domande.

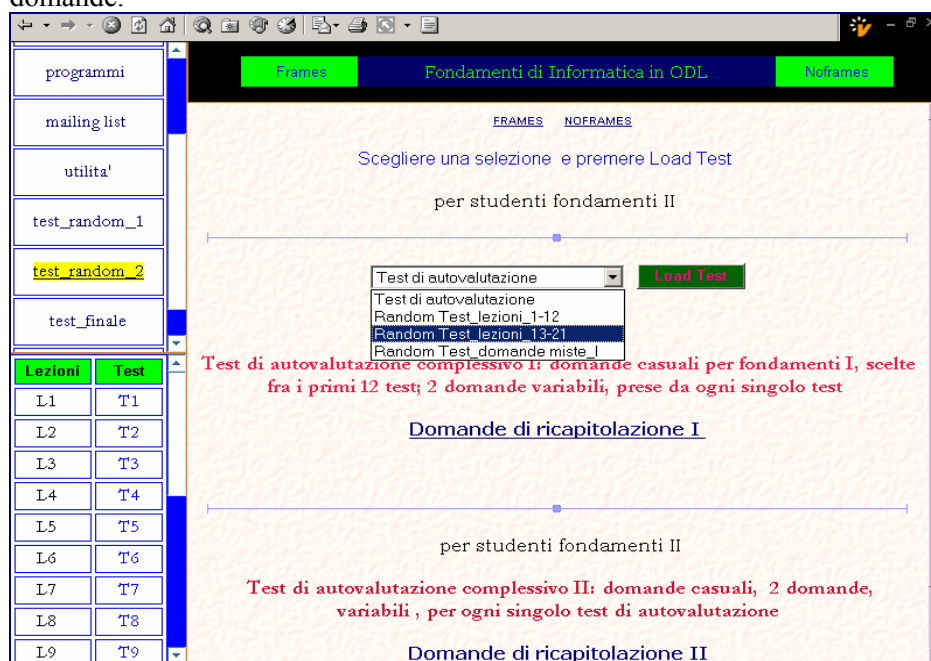


Fig. 3.14 - (Videata dei links ai test random F2)

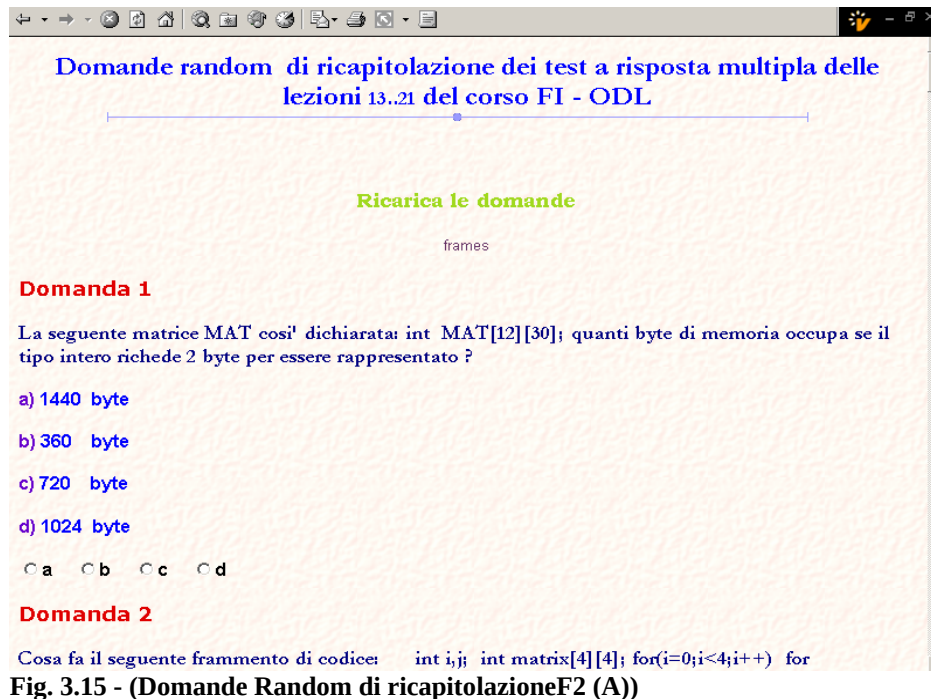


Fig. 3.15 - (Domande Random di ricapitolazioneF2 (A))

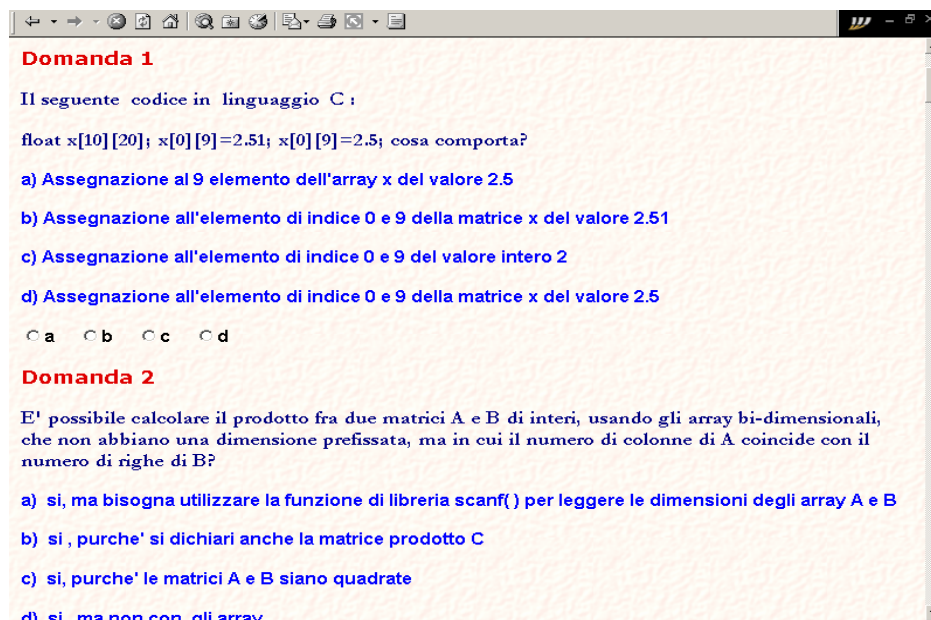
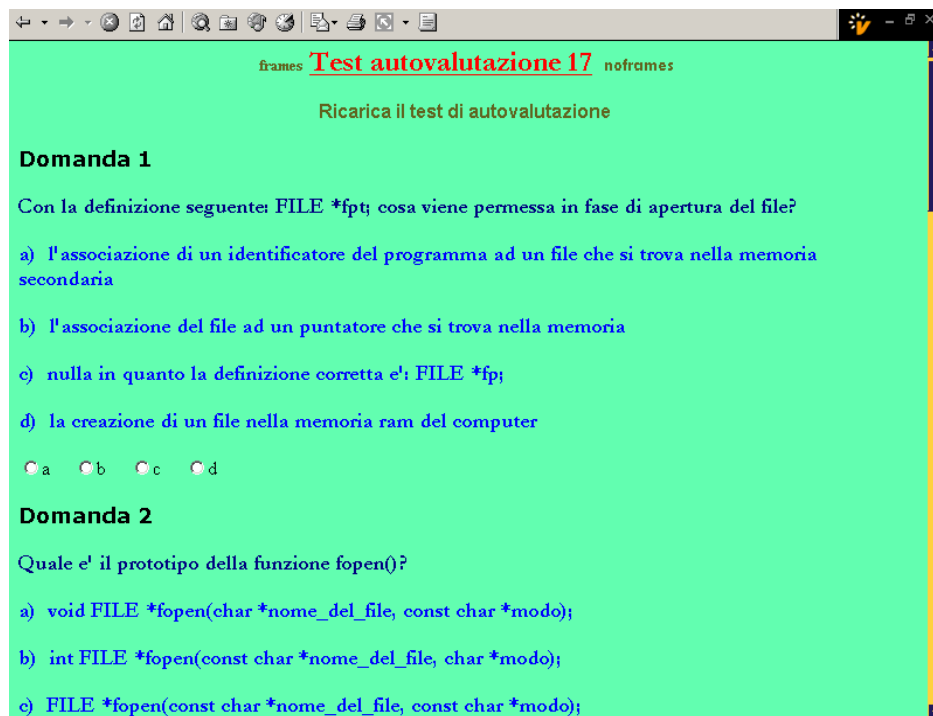


Fig. 3.16 - (Domande Random di ricapitolazione F2 (B))

Cliccando su **ricarica le domande** in fig.3.15 appare nel sito una schermata simile alla fig. 3.15 in cui le domande però sono cambiate in modo random (casuale secondo un prefissato algoritmo) come visibile nella successiva figura 3.16.

Esempio di Test di autovalutazione a domande variabili TEST 17



frames **Test autovalutazione 17** noframes

Ricarica il test di autovalutazione

Domanda 1

Con la definizione seguente: FILE *fpt; cosa viene permessa in fase di apertura del file?

- a) l'associazione di un identificatore del programma ad un file che si trova nella memoria secondaria
- b) l'associazione del file ad un puntatore che si trova nella memoria
- c) nulla in quanto la definizione corretta e': FILE *fp;
- d) la creazione di un file nella memoria ram del computer

☐ a ☐ b ☐ c ☐ d

Domanda 2

Quale e' il prototipo della funzione fopen()?

- a) void FILE *fopen(char *nome_del_file, const char *modo);
- b) int FILE *fopen(const char *nome_del_file, char *modo);
- c) FILE *fopen(const char *nome_del_file, const char *modo);

Fig. 3.17 - (Domande Random Test 17 (A))

Si può notare, nella figura 3.18, come le domande contenute nel Test di autovalutazione 17, mostrato in figura 3.17, cambino in modo casuale ogni volta che il test viene caricato.

Esse venivano scelte randomicamente all'interno di un archivio di domande didatticamente omogenee.

Anche per i test di autovalutazione random era fornito sia il voto, in trentesimi, riportato dai lerners sia il tempo totale impiegato per rispondere alle domande.

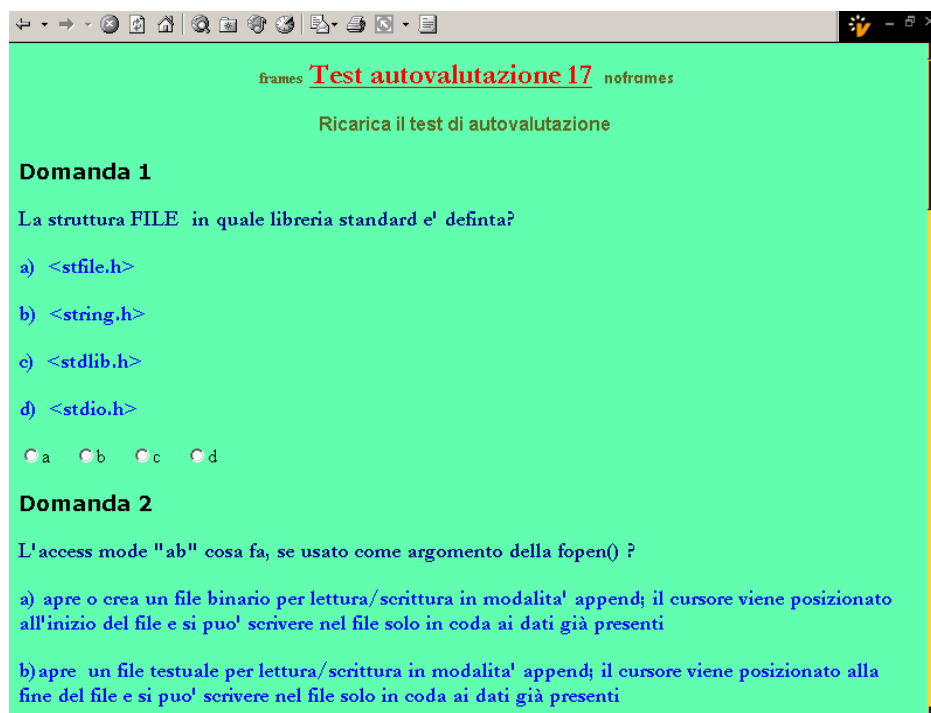


Fig. 3.18 - (Domande random Test 17 (B))

3.6 Area Riservata

Alle cartelle di tutti i protagonisti del Corso FIODL si accedeva dall'area riservata mostrata in Figura 3.19.

Docente e tutor avevamo anche noi una cartella personale che riportava l'andamento di tutti i learners dei corsi.

Anche le nostre cartelle erano protette da password e riportavano i voti di tutti gli studenti remoti, le loro password e tutte le soluzioni delle prove intermedie proposte.

Tali cartelle fungevano quindi anche da interfaccia comunicativa fra tutor e docente. In figura 3.20 mostriamo un esempio del contenuto *cartella docente*.

I learners dall'area riservata, accedevano alle cartelle personali tramite uno *user name* ed una *password* personali.

Nelle cartelle, come detto, potevano prendere visione, (*funzione di bacheca personale*), delle soluzioni proposte alle prove intermedie da loro

affrontate, del voto in esse riportato e di eventuali consigli su cosa insistere nello studio (visione del proprio rating).

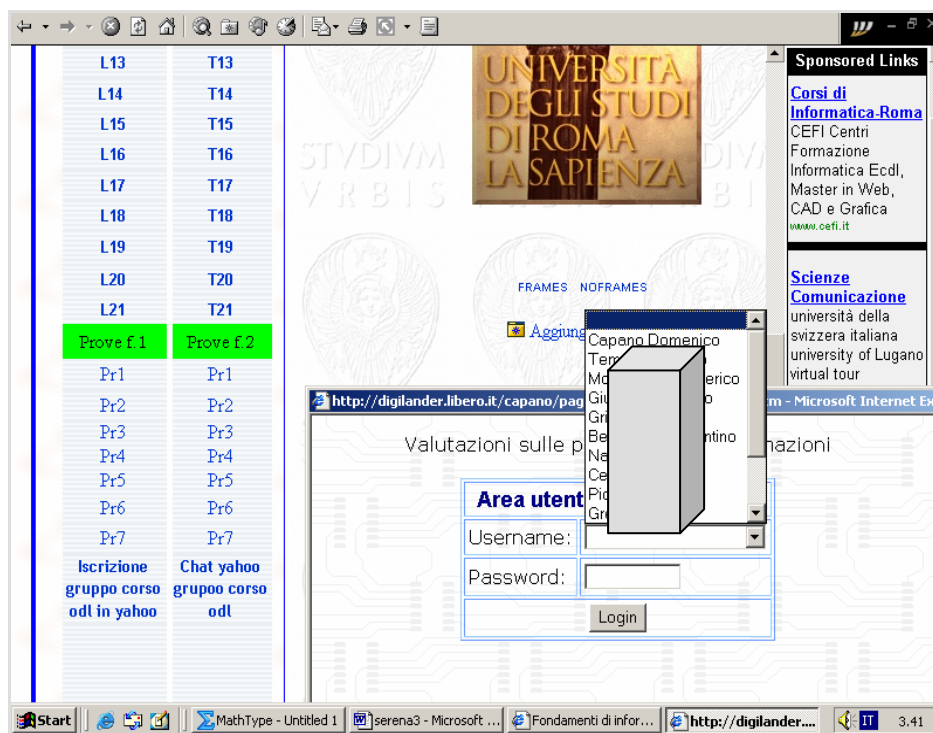


Fig. 3.19 - (Area riservata utenti FIODL)

Nella figura seguente mostriamo una videata del contenuto della cartella personale del docente del corso FIODL.

Sul lato destro è visualizzata parte della soluzione per F2 della Prova Intermedia 7 ottenibile cliccando, sul link **Sol.7**, nella figura in basso.

Similmente avviene per le altre prove intermedie e loro soluzioni nonché per il test introduttivo e per il test finale sia per F1 che F2.

Il contenuto delle varie cartelle personali e pubbliche, dei protagonisti di un corso ODL, deve essere aggiornabile online; per questo le piattaforme LMS e LCMS, oggi, utilizzano linguaggi server-side tipo PHP ed ASP (vbscript) che usano rispettivamente database MySQL ed Access per la gestione delle cartelle.

PHP e MySQL sono le soluzioni preferite, dagli implementatori di piattaforme, per la loro più alta efficienza, quando il numero di utenti cresce. Anche ogni docente, nei modelli di e-Learning centrati sull'utente, ha la sua

cartella personale indipendente dai corsi la quale contiene i materiali didattici dei corsi (dispense, esercizi, slide), oltre che tutte le informazioni, di interesse didattico, sui discenti.

The screenshot displays a web application interface. On the left, a sidebar contains a navigation menu with items like 'L20', 'L21', 'Prove f1', 'Pr1', 'Pr2', 'Pr3', 'Pr4', 'Pr5', 'Pr6', 'Pr7', 'Iscrizione gruppo corso', and 'odt in yahoo'. The main content area is divided into two sections. The top section is a table titled 'Test introduttivo' and 'Test finale' showing scores for various students. The bottom section is titled 'Prove intermedie e loro soluzioni' and contains two sub-tables for 'fondamenti 1' and 'fondamenti 2'. The right sidebar shows the 'Soluzione prova intermedia 7 per fondamenti 2' with a C program solution.

Cognome nome	Test introduttivo	PR.1	PR.2	PR.3	PR.4	PR.5	PR.6	PR.7	Test finale	password
M. Boni	13/30	.	27/30	.	.	29/30	.	27/30	.	gos
F. Boni	11/30	28/30	26/30	.	.	.	.	.	.	ada
C. Ar	5/30	29/30	30/30	29/30	29/30	30/30	29/30	19/30	.	gin
B. C.	3/30	25/30	17/30	19/30	18/30	19/30	21/30	21/30	16/30	ulus
D. F.	8/30	.	.	.	.	.	.	.	.	lis
N. S.	10/30	29/30	f_n_s	28/30	f_n_s	28/30	25/30	30/30	18/30	eus
G. R.	9/30	29/30	17/30	25/30	28/30	.	.	.	.	ust
P. R.	7/30	29/30	f_n_s	30/30	f_n_s	30/30	28/30	27/30	24/30	sis
P. F.	9/30	27/30	f_n_s	29/30	f_n_s	29/30	29/30	30/30	22/30	ous
B. M.	12/30	26/30	f_n_s	29/30	f_n_s	.	.	.	.	tos
F. V.	12/30	.	.	.	.	.	.	.	.	ntus
C. D.	10/30	30/30	f_n_s	30/30	f_n_s	30/30	29/30	27/30	24/30	ilva

	test.1	pr.1	pr.2	pr.3	pr.4	pr.5	pr.6	pr.7	test 1 finale
fondamenti 1	sol.1	sol.2	sol.3	sol.4	sol.5	sol.6	sol.7		

	test.2	pr.1	pr.2	pr.3	pr.4	pr.5	pr.6	pr.7	test 2 finale
fondamenti 2	sol.1	sol.2	sol.3	sol.4	sol.5	sol.6	sol.7		

```

/* 1) Scrivete un programma in
linguaggio C, il quale concateni due
stringhe
lette da input, di al piu' 35 caratteri
ciascuna; le funzioni utilizzate
devono ricevere, come parametri,
gli array considerandoli puntatori;
la scansione degli array deve
avvenire mediante espressioni
puntatore.*/

#include<stdio.h>
#define L_MAX 35
void leggi_stringa(char
*);
void concatena(char
*,char *, char *);

int main()
{
char car;
char prima_stringa
[L_MAX+1],
seconda_stringa[L_MAX+1];
char terza_stringa
[2*L_MAX+1];
int i=0,j=0;
leggi_stringa
(prima_stringa);
leggi_stringa
(seconda_stringa);
printf("\nho letto: %s e
%s",prima_stringa,seconda_s

```

Fig. 3.20 - (Esempio di contenuto della cartella personale del docente)

3.7 Test di valutazione finale

Nelle sottostanti figure 3.21 mostriamo parte del Test finale (o Prova finale) del corso FIODL predisposto per gli studenti di Fondamenti 2 disponibile nel sito web .

test finale II - Microsoft Internet Explorer fornito da Virgilio

Test finale del corso di FI-ODL

"Per gli studenti di fondamenti di informatica II "

Questo test serve per valutare il livello finale della preparazione raggiunta dai partecipanti al corso, al fine di analizzare se il corso e' stato utile.

Buon lavoro.

Compilate il questionario in tutte le sue parti ed inviate

Cognome :

Nome :

Num. di matricola :

E-mail :

Corso :

Start | 1. Radi... | Tesifina... | HEADE... | test fin... | 2.21

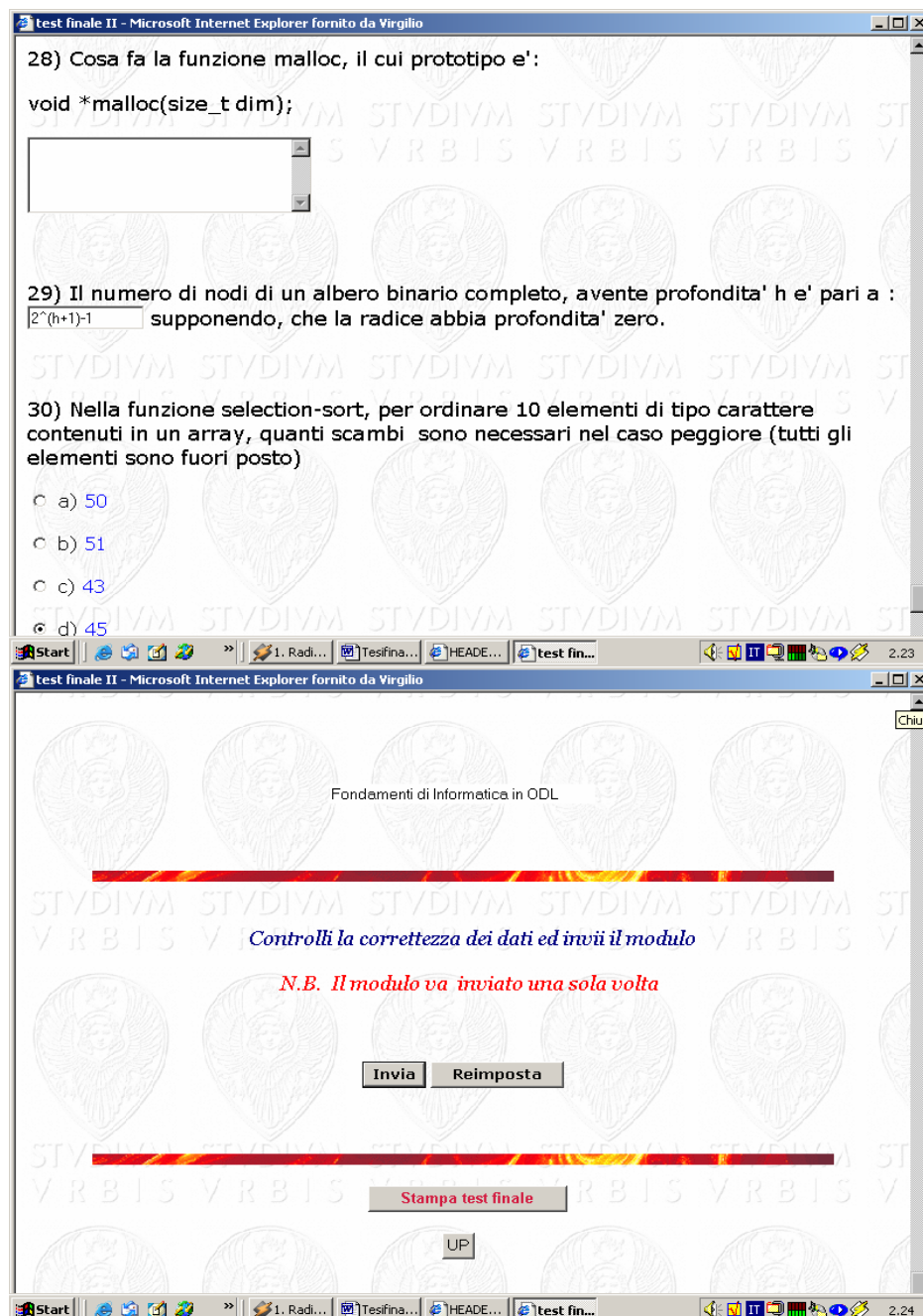


Figure 3.21 - (Prova Finale Fondamenti 2)

Nell'appendice A riportiamo integralmente la medesima Prova Finale (o Test Finale).

Questa prova è stata affrontata dai learners, su invito dei tutors, in contemporanea onde simulare l'esame finale in presenza.

In tal caso si è persa, naturalmente, l'asincronicità del corso a distanza, ma ciò era necessario per limitare l'interrelazione fra i learners, non corretta se si voleva avere un metro di giudizio più obiettivo sulla singola preparazione acquisita.

In genere nell'e-Learning odierno si adottano soluzioni di tipo misto *blended e-Learning*, facendo comprendere, fra la componente didattica in presenza, sicuramente la prova finale.

A soffrire della perdita di asincronicità del corso è il learning collaborativo fra studenti e studenti, come abbiamo potuto notare dai dati emersi nella valutazione del corso⁵⁰, soluzione auspicata nella didattica a distanza moderna basata sul *costruttivismo o costruzionismo*⁵¹.

Il learning collaborativo, nel nostro esperimento, ha sofferto anche per via del fatto che abbiamo invitato i discenti a non fornire, ai colleghi di corso, le loro soluzioni delle prove intermedie essendo queste ultime svolte in modo asincrono dai discenti.

Dalla nostra esperienza è emerso che, per incentivare il learning collaborativo ed al contempo valutare i progressi fatti dal singolo discente, sia necessario adottare soluzioni didattiche sincrone anche nelle prove intermedie⁵² e ciò è poco desiderabile, poco praticabile e più costoso.

Da non dimenticare però che le prove intermedie hanno il fine, anche, di tenere desta l'attenzione del discente remoto che potrebbe sentirsi isolato in un corso a distanza; isolamento causa di un possibile abbandono del corso.

I risultati forniti sia dalla Prova Finale per F2, sia per F1 sono disponibili nel successivo capitolo 4.

⁵⁰ Vedi sottosezione 7.1.1

⁵¹ Le tre grandi teorie psicopedagogiche del novecento sono *comportamentismo*, *cognitivismo* e *costruttivismo* da cui piattaforme e strategie di e-Learning hanno derivazione pedagogica. Il costruttivismo ha fra le sue caratteristiche più salienti quella di porre il *discente al centro del processo educativo*; dà particolare importanza al contesto educativo ed alla collaborazione fra discenti e formatori, il cui ruolo si va ridimensionando in termini di "centro del sapere" in misura proporzionale alla capacità dei formatori di creare comunità di apprendimento "learning community" paritarie e condivise.

⁵² Ricordiamo che nel nostro esperimento le prove intermedie originano i crediti formativi intermedi del corso per il singolo discente.

4 Risultati prodotti dal corso

4.1 Grafici di analisi dei risultati forniti dall'esperimento

Nel Corso FIODL sono state assegnate sette prove intermedie per il corso1,1 e sette prove intermedie per il corso1,2 con lo scopo di tenere desta l'attenzione del discente e diminuire il senso di isolamento tipico dei corsi a distanza e poter assegnare i crediti formativi.

Per il corso1,1, sei prove avevano il vincolo della "obbligatorietà" ed una prova era facoltativa. Dei sei studenti iscritti al livello Fondamenti 1 quattro studenti hanno riportato in media i voti presenti nella figura 4.1.

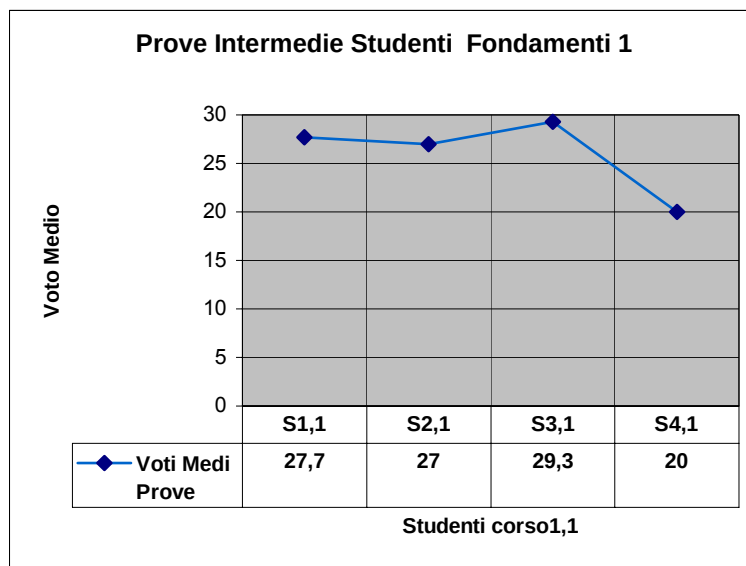


Fig. 4.1 - (Voti Medi Prove F1 (media semplice))

Nel corso sono state assegnate agli studenti iscritti al corso1,2, Fondamenti di Informatica 2, altre sette prove intermedie anch'esse con lo scopo di tenere desta l'attenzione del discente e diminuire il senso di isolamento tipico dei corsi a distanza e poter assegnare i crediti formativi di corso.

Cinque prove intermedie avevano il vincolo della "obbligatorietà" e due prove erano facoltative. Dei sei studenti iscritti al livello Fondamenti 2 sei studenti hanno riportato in media i voti presenti nella figura 4.2.

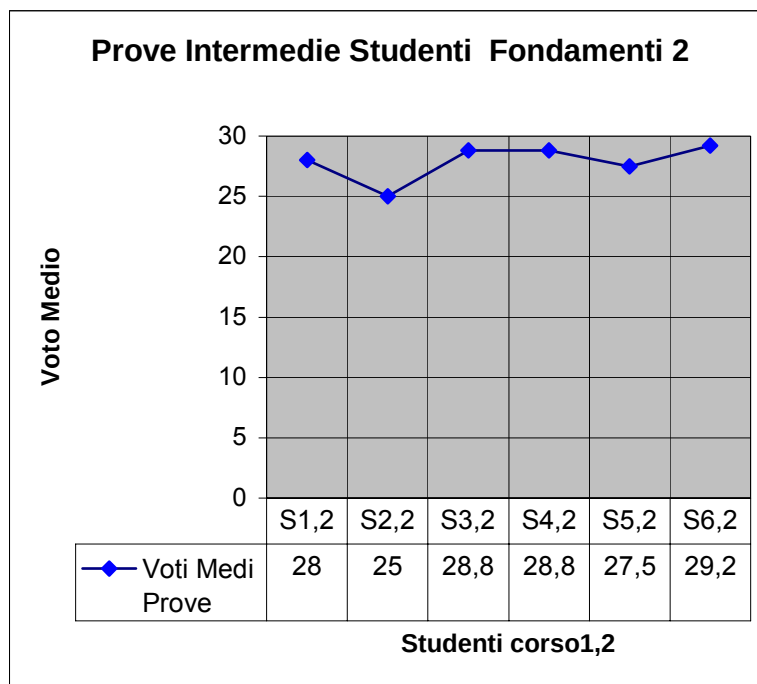


Fig. 4.2 - (Voti Medi Prove F2 (media semplice))

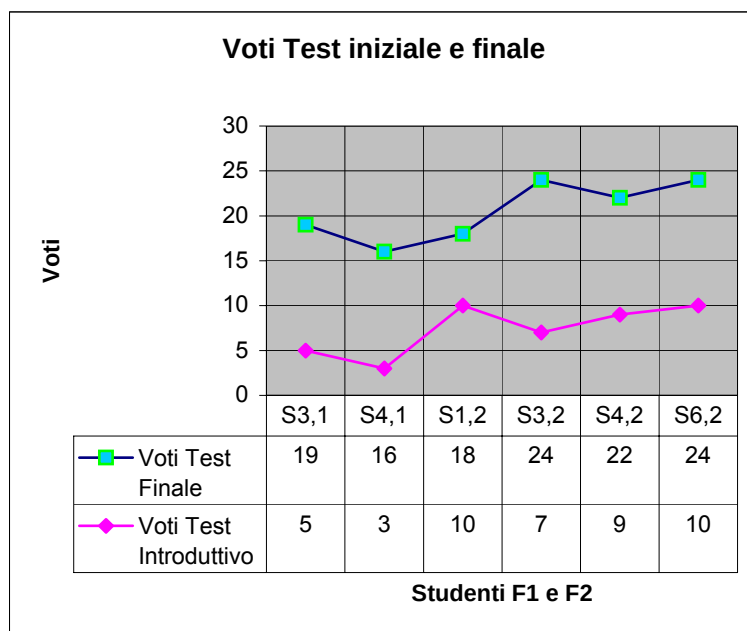


Fig. 4.3 - (Voti Test Introduttivo e Prova Finale)

Nella figura 4.3 mostriamo i voti riportati dagli studenti che hanno affrontato sia il Test Introduttivo sia la Prova Finale; si può notare che l'area, al di sotto della spezzata Voti Test Introduttivo, rappresenta la *conoscenza iniziale* che i learners avevano degli argomenti trattati nel corso a loro corrispondente (corso1,1 o corso1,2).

L'area al di sotto della spezzata Voti Test Finale, rappresenta la *conoscenza finale* degli argomenti trattati nel corso a loro corrispondente. L'area compresa fra le due spezzate su citate può essere interpretata come la *conoscenza acquisita* nel corso a loro corrispondente.

Nella seguente figura 4.4 si è aggiunta rispetto alla figura 4.3 la corrispondente media semplice riportata dai suddetti studenti.

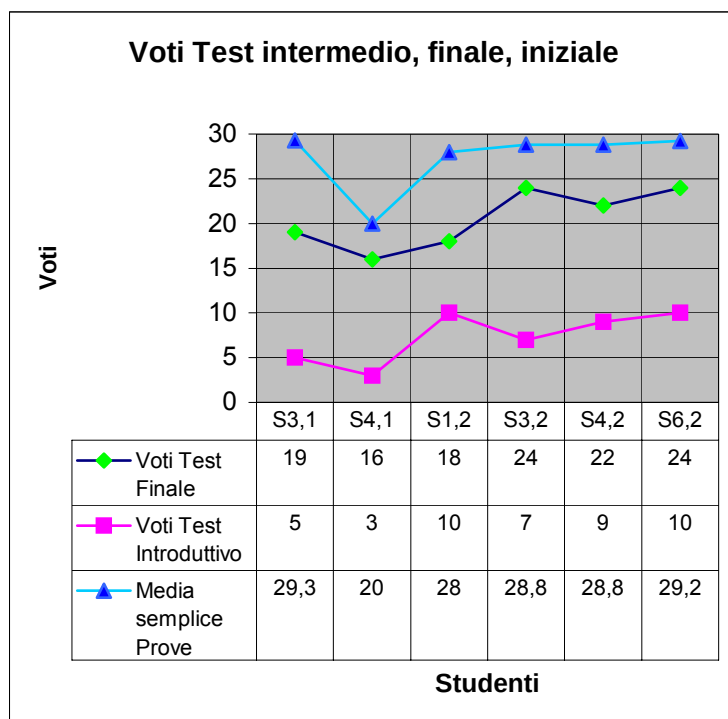


Fig. 4.4 - (Voti Test Finale, Introd. e Prove intermedie)

4.2 Generazione crediti formativi in FIODL

Inseriremo adesso i grafici con i crediti formativi, acquisiti dai learners nelle Prove Intermedie, applicando la, già vista, metodologia di generazione della sottosezione 1.3.3.

A tal fine introduciamo alcune considerazioni preliminari.

Considerazione a)

Le Prove Intermedie, aventi carattere facoltativo, generano un credito formativo solo se affrontate positivamente, mentre non generano alcun debito formativo se non affrontate od affrontate negativamente.

Considerazione b)

Le Prove Intermedie se non sono erogate alla fine di un Modulo, per scelta dei tutors e/o docenti, debbono ugualmente essere etichettate.

Considerazione c)

La Prova Intermedia è un SottoModulo contenuto nel Modulo cui s'intende indagare/autovalutare la preparazione acquisita dai learners.

Considerazione d)

La generica Prova Intermedia $PI_{c, j_c, m_c j_c}$, che abbiamo detto essere un SottoModulo del generico Modulo $Mod_{c, j_c, m_c j_c}$, essendo collocata all'interno di tale Modulo alla sua fine (erogata dopo l'ultimo test di autovalutazione), ha indice di sottomodulo pari al numero di SottoModuli

contenuti in tale Modulo che abbiamo chiamato $L_{c j_c m_c j_c}$ ⁵³, ossia è il

SottoModulo $SMod_{c, j_c, m_c j_c, L_{c j_c m_c j_c}}$.

Considerazione e)

Il *peso delle Prove Intermedie* lo riferiamo all'indice del Modulo e non del SottoModulo per via del fatto che abbiamo scelto di inserire una sola Prova Intermedia per Modulo. Il peso rappresenta, quindi, un indice della difficoltà didattica del Modulo in questione.

⁵³ Vedere sottosezione 1.3.1

Ciò detto inseriamo due tabelle (4.1 e 4.3) per i due gruppi di learners (un gruppo di Fondamenti 1 ed un gruppo di F2), riportanti il voto ricevuto nelle Prove Intermedie, il loro Peso, il Modulo a cui le prove appartengono ed il Vincolo (caratteristica) della prova intermedia.

Tab. 4.1 - Moduli e SottoModuli di tipo Prove Intermedie F1

Per learners Fondamenti di Informatica 1				
Prova Int.	1	-	2	3
Modulo	Mod1,1,2	Mod1,1,3	Mod1,1,4	Mod1,1,5
SottoModulo	SMod1,1,2,11	SMod1,1,3,3	SMod1,1,4,3	SMod1,1,5,7
Peso	PesoPI1,1,2	-	PesoPI1,1,4	PesoPI1,1,5
Voto k	Voto 1	-	Voto 2	Voto 3
Vincolo	Obbligatoria	Non erogata	Facoltativa	Obbligatoria
Prova Int.	4	5	6	7
Modulo	Mod1,1,6	Mod1,1,7	Mod1,1,8	Mod1,1,9
SottoModulo	SMod1,1,6,3	SMod1,1,7,3	SMod1,1,8,5	SMod1,1,9,3
Peso	PesoPI1,1,6	PesoPI1,1,7	PesoPI1,1,8	PesoPI1,1,9
Voto k	Voto 4	Voto 5	Voto 6	Voto 7
Vincolo	Obbligatoria	Obbligatoria	Obbligatoria	Obbligatoria

Nella Tabella 4.1 possiamo notare, ad esempio, che la Prova Intermedia numero 3 è il SottoModulo SMod1,1,5,7 del Modulo Mod1,1,5 ed il numero di sottomoduli contenuti in Mod1,1,5 è $L_{115}=7$.

In tale modulo vi sono, quindi, tre SottoModuli di tipo Lezione e tre SottoModuli di tipo Test di autovalutazione prima di essa (come si può vedere guardando la Tabella 1.4).

E' una Prova Intermedia con il vincolo dell'obbligatorietà, e con un peso pari a PesoPI1,1,5, in cui i learners che l'affrontano sono giudicati/autovalutati attraverso la misura numerica intera Voto 3.

Particolarizziamo la Tabella 4.1 al caso delle Prove Intermedie affrontate dal learner S4,1 (vedere la tabella 4.2), tenendo in considerazione che tutto,

della Tabella 4.1 è valido in generale per gli studenti di fondamenti 1, tranne la riga Voto k che in generale può essere diversa per ogni learner.

Tab. 4.2 - Voti Prove Intermedie del Learner S4,1

Prova Int.	1	-	2	3	4	5	6	7
Modulo	..	..	..	..	..	..	..	..
SottoModulo	..	..	..	..	..	..	..	..
Peso	1	-	3	2	3	1	2	3
Voto k	25	-	17	19	18	19	21	21
Vincolo	Ob.	Non e.	Fac.	Ob.	Ob.	Ob.	Ob.	Ob.

Adesso si può procedere al calcolo dei crediti formativi acquisiti dal learner S4,1 utilizzando le formule da (1.14) a (1.19). Assegniamo a **Maxcredito**_{1,1} il valore 6 punti. Applicando la formula (1.14), della sottosezione 1.3.3, seguente avremo i Voti della Prove Intermedie pesate svolte positivamente:

$$VPIp_{c,j_c,m_c,j_c^s} = PesoPI_{c,j_c,m_c,j_c^s} * Voto_s / 30$$

$$VPIp_{1,1,2,1} = 1*25/30; VPIp_{1,1,5,3} = 2*19/30; VPIp_{1,1,6,4} = 3*18/30;$$

$$VPIp_{1,1,7,5} = 1*19/30; VPIp_{1,1,8,6} = 2*21/30; VPIp_{1,1,9,7} = 3*21/30.$$

Con la (1.15) seguente:

$$PesoVotoPI^+_{c,j_c} = \frac{\sum_{s=1}^{NPI\ svolte} VPIp_{c,j_c,m_c,j_c^s}}{\sum_{k=1}^{NPI\ totali} PesoPI_{c,j_c,m_c,j_c^k}}$$

Ricaviamo

$$PesoVotoPI^+_{1,1} = \frac{1*25/30 + 2*19/30 + 3*18/30 + 1*19/30 + 2*21/30 + 3*21/30}{1+2+3+1+2+3}$$

Ossia **PesoVotoPI** + 1,1 = 6,766/12 = 0,5638

Credito1,1risultante = $\lceil 0,5638 \cdot 6 \rceil = + 4$; Debito1,1risultante = 0; quindi **Creditiassegnati** = + 4.

Allo stesso modo si procede con gli altri learners di Fondamenti 1 ed otteniamo per il learner S3,1 Creditiassegnati = + 6.

Tab. 4.3 - Moduli e SottoModuli di tipo Prove Intermedie F2

Per learners Fondamenti di Informatica 2				
Prova Int.	1	2	3	4
Modulo	Mod1,2,2	Mod1,2,3	Mod1,2,4	Mod1,2,5
SottoModulo	SMod1,2,2,3	SMod1,2,3,5	SMod1,2,4,3	SMod1,2,5,3
Peso	PesoPI1,2,2	PesoPI1,2,3	PesoPI1,2,4	PesoPI1,2,5
Voto k	Voto 1	Voto 2	Voto 3	Voto 4
Vincolo	Obbligatoria	Facoltativa	Obbligatoria	Facoltativa
Prova Int.	5	6	7	
Modulo	Mod1,2,6	Mod1,2,7	Mod1,2,8	
SottoModulo	SMod1,2,6,3	SMod1,2,7,3	SMod1,2,8,5	
Peso	PesoPI1,2,6	PesoPI1,2,7	PesoPI1,2,8	
Voto k	Voto 5	Voto 6	Voto 7	
Vincolo	Obbligatoria	Obbligatoria	Obbligatoria	

Particolarizziamo la Tabella 4.3 al caso delle Prove Intermedie affrontate dal learner S6,2, tenendo in considerazione che tutto della Tabella 4.3 è valido in generale per gli studenti di fondamenti 2 tranne la riga Voto k

Tab. 4.4 - Voti Prove Intermedie del Learner S6,2

Prova Int.	1	2	3	4	5	6	7
Modulo	..	..	..	..	..	..	..
SottoModulo	..	..	..	..	..	..	..
Peso	1	1	2	1	2	2	3
Voto k	30	x	30	x	30	29	27
Vincolo	Ob.	Fac.	Ob.	Fac.	Ob.	Ob.	Ob.

Utilizziamo il medesimo procedimento, visto sopra, per il calcolo dei crediti formativi acquisiti dal learner S6,2 utilizzando le formule da (1.14) a (1.19). Assegniamo a **Maxcredito**_{1,2} il valore 6 punti.

Applicando la formula (1.14), della sottosezione 1.3.3, avremo:

VPIp_{1,2,2} **1**= 1*30/30; **VPIp**_{1,2,4} **3**= 1*30/30 **VPIp**_{1,2,5} **5**= 2*30/30;

VPIp_{1,2,7} **6**= 2*29/30; **VPIp**_{1,2,8} **7**=3*27/30.

$$PesoVotoPI^+_{1,2} = \frac{1*30/30+2*30/30+ 2*30/30+2*29/30+3*27/30}{1+2+2+2+3}$$

PesoVotoPI_{1,2}= 9,633/10 =0,9633;

il credito Risultante per il learner S6,2 è il seguente:

$$Credito_{risultanteS6,2} = \lceil 0,9633*6 \rceil = \lceil 5,78 \rceil = +6$$

Il Debito risultante è uguale a zero quindi: **Crediti assegnati**= +6 punti.

Similmente si possono calcolare i crediti formativi per gli altri learners di Fondamenti 2; in figura 4.5 descriviamo l'andamento grafico dei crediti formativi di tutti i learners che hanno partecipato anche alla Prova Finale.

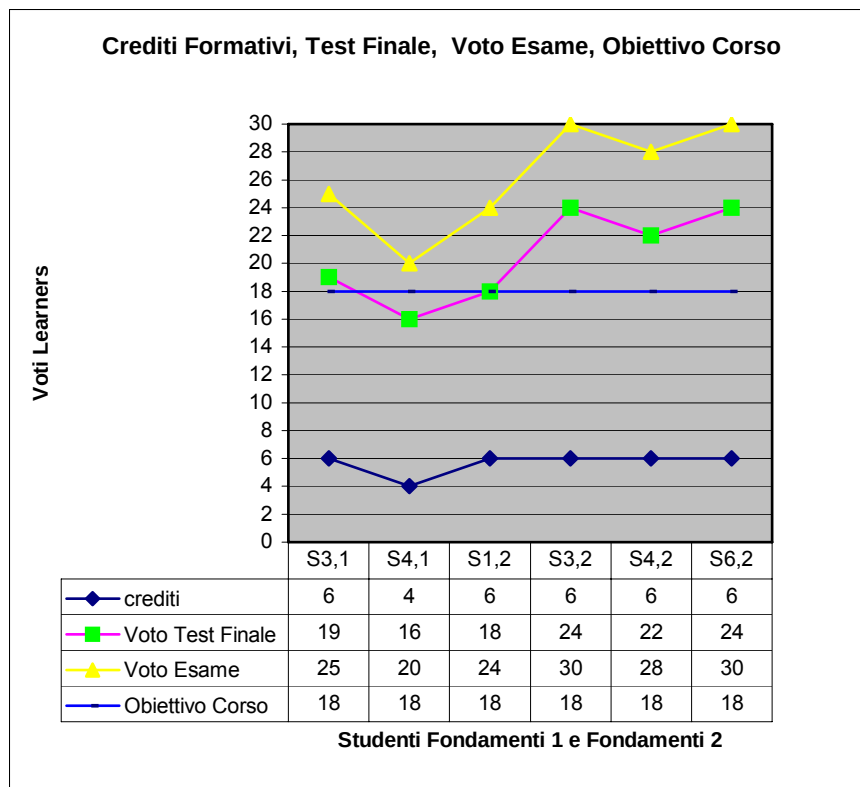


Fig. 4.5 - (Crediti formativi, Voti Test Finale e Voto Esame)

Analizzando la figura 4.5 si può notare che l'*obiettivo del corso* è stato raggiunto da tutti i learners del corso1,2 che hanno affrontato la Prova Finale, mentre nel corso1,1 un learner ha raggiunto l'obiettivo del corso mentre il learner S4,1 si è fermato sotto la soglia del 18.

Se considerassimo il Voto Esame come parametro di verifica dell'obiettivo del corso e quindi come "misura" dell'apprendimento dei learners, notiamo dalla figura che l'obiettivo è stato ampiamente raggiunto da tutti i learners, con due learners che hanno raggiunto l'obiettivo massimo Voto Esame pari a 30.⁵⁴

⁵⁴ L'analisi statistica riportata in questo capitolo risente, naturalmente, del limite dell'esiguo numero di discenti campionati; comunque i metodi utilizzati sono generali e congrui con il fine didattico di questo testo.

4.3 Risultati del Questionario di valutazione del corso FIODL

Riportiamo nella tabella 4.5 le risposte in forma aggregata al questionario di valutazione del corso inviate, tramite e-mail, dagli studenti del corso^{1,1} e corso^{1,2}.

Il Questionario di valutazione del corso, da parte dei learners che vi hanno partecipato, con le domande è consultabile nell'Appendice B.

Tab. 4.5 - Valutazione corso learners F1 e F2

Domanda	Opzione 1	Opzione 2	Opzione 3	Opzione 4	Opzione 5
1	Sperimentare l'ODL 50%	Interesse per il corso 50%	Interesse materiale aggiuntivo 83%	Altri Motivi 50%	
2	Molto Alta 17%	Alta 83%	Sufficiente 0%	Scarsa 0%	
3	Molto Buona 17%	Buona 83%	Sufficiente 0%	Insufficiente 0%	Mediocre 0%
4	Molto 67%	Sufficientemente 17%	Scarsamente 16%		
5	Troppo Lunga 0%	Della Giusta Lunghezza 100%	Troppo Corta 0%	Altri Commenti 0%	
6	Molto Buoni 50%	Buoni 50%	Sufficienti 0%	Insufficienti 0%	Mediocri 0%
7	Difficili ma Autoistruttive 33%	Difficili 0%	Di Normale Difficoltà 67%	Facili 0%	
8	Ottime 50%	Più che Sufficienti 33%	Sufficienti 17%	Insufficienti 0%	
9	Troppo Veloci 0%	Veloci 67%	Sufficientemente Veloci 33%	Scarsamente Veloci 0%	
10	Più volte al giorno 50%	Una volta al giorno 0%	Una volta ogni due giorni 17%	Settimanale 33%	Altro 0%
11	Docenti 67%	Colleghi 0%	Nessuno 33%	Altri 0%	

12	Più che Sufficiente 50%	Sufficiente 50%	Scarso 0%		
13	Più che Sufficiente 17%	Sufficiente 33%	Scarso 50%		
14	Si, Molto 17%	Si, Un pò 16%	No 67%		
15	Si 67%	Abbastanza 33%	No 0%		
16	Si 33%	No 17%	Non vedo sostanziali differenze 16%	Non so 33%	
17	Seguire le lezioni da casa 17%	Scegliere momento per studiare 67%	Nessun vantaggio 0%	Altro 0%	
18	Sempre 33%	In molti casi 67%	In pochi casi 0%	Mai 0%	
19	Quasi tutti 75%	Metodo tradizionale 25%			
20	Si 100%	No 0%	Non so 0%		
21	Migliore 50%	Uguale 50%	Inferiore 0%		
22	Molto Buono 67%	Buono 33%	Sufficiente 0%	Insufficiente 0%	
23	Si 100%	No 0%	Non so 0%		
24	Voto medio assegnato a FIODL 27				

Analizzando i dati della tabella 4.5 possiamo fare alcune considerazioni per grandi linee.

La motivazione dei learners, per la didattica a distanza, prima di iniziare i corsi era alta, la qualità didattica dei corsi è stata giudicata positivamente e molto utili sono stati considerati i test di autovalutazione e le prove intermedie; i tutors sono stati giudicati presenti nel soddisfare le loro richieste didattiche; il learning collaborativo fra studenti è l'unica nota dolente anche se giustificato parzialmente dalla presenza delle prove di valutazione intermedie (interazione fra docente/tutor e discenti); nel complesso FIODL è stato considerato migliore dei corsi tradizionali forniti da NETTUNO al tempo

(lezioni in videocassette) per via della maggiore interazione che qui gli studenti hanno avuto con i docenti, tutors e didattica; il voto complessivo medio assegnato dai learners a FIODL è stato 27/30.

Il processo con cui abbiamo realizzato il corso FIODL, come abbiamo potuto notare nelle pagine precedenti, è simile a quello dei corsi realizzati oggi in modalità e-Learning, basato sul seguente modello (detto *modello ADDIE*: **A**nalysis, **D**esign, **D**evelopment, **I**mplementazione ed **E**valuation):

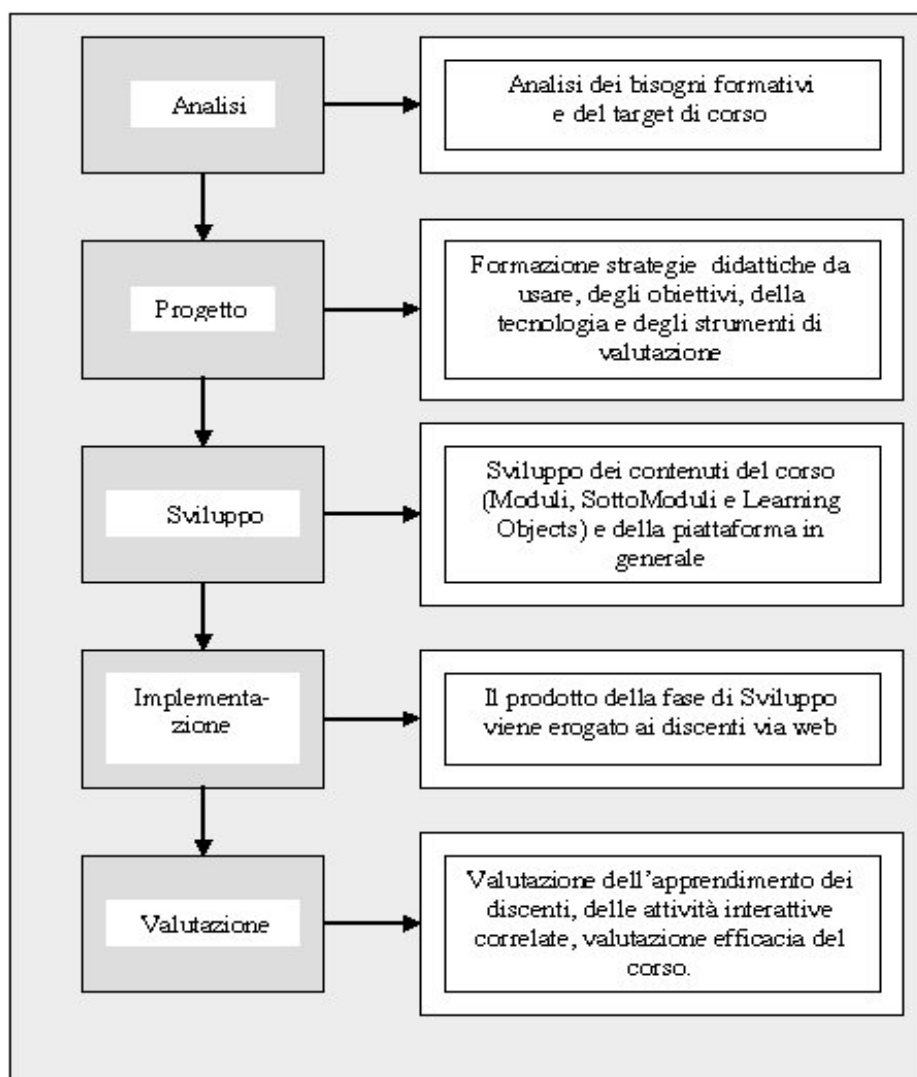


Fig. 4.6 - (Modello ADDIE)

Il Modello di ADDIE è un modello sequenziale, con le fasi costituenti che debbono eseguirsi una dopo l'altra in modo rigoroso e quindi risulta essere scarsamente flessibile; ha il vantaggio in ogni modo di rendere sistematico lo sviluppo di un progetto di e-Learning ma può richiedere tempi di sviluppo eccessivamente lunghi oltre che presentare un'eccessiva inflessibilità.

Altri modelli cercano di superare i limiti del modello ADDIE e sono i cosiddetti *design iterativi*.

In questi modelli i corsi, sono costruiti in team e rapidamente sottoposti al giudizio critico dei discenti (fase di test utente); i discenti quindi sono parte integrante del processo di design.

Con tale approccio progettuale non vi è però una distinzione chiara fra le varie fasi ed inoltre non si può prevedere quante revisioni debbono essere fatte prima di sottoporre al mercato un prodotto avente un accettabile grado qualitativo.

Oggi è prevalentemente utilizzato il modello di ADDIE Modificato (**Modified ADDIE Model**), compromesso fra i due modelli citati, in cui è stata introdotta una nuova fase, chiamata **Prototyping fase**, compresa fra la fase di Design e la fase Development.

Nella fase Prototyping i discenti valutano un prototipo del corso; verificano principalmente la compatibilità con la piattaforma esistente ed altri difetti eventuali; il *feedback dei discenti* ottenuto nella fase di Prototyping è sfruttato immediatamente, per risolvere le anomalie del corso, nella fase di design; poi un nuovo corso con le correzioni apportate è proposto ai discenti.

Dopo un congruo numero di cicli si arriva al prodotto con una valutazione positiva da parte dei discenti e quindi si passa alla fase successiva **Implementation**.

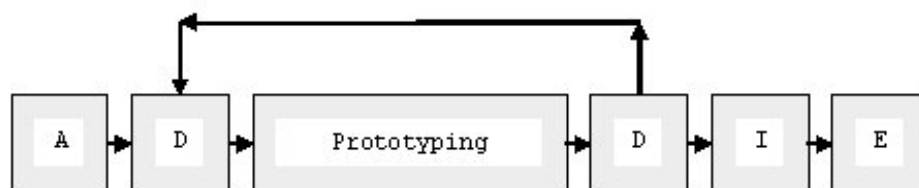


Fig. 4.7 - (Modified ADDIE Model)

5 Come puntare il focus sull'apprendimento

5.1 Path learning

In questo capitolo tracciamo alcune linee guida, per qualche lettore che volesse approfondire l'argomento su un possibile metodo d'automatizzazione di un corso on-line che preveda il riassettaggio delle componenti di un Corsoc.jc in base al feedback (col fine di minimizzare il loro gap conoscitivo⁵⁵) ricevuto nelle varie prove in cui si cimentano i learners.

Analizziamo prima gli obiettivi formativi di un Modulo, di un SottoModulo e dei Learning Objects. I Learning Objects, di per se autoconsistenti, sono aggregati fra loro al fine di creare un SottoModulo che dev'essere completo in termini contenutistici, funzionali e formativi in funzione di specifici obiettivi didattici, finalizzati al raggiungimento dell'obiettivo didattico del Modulo. I Moduli sono aggregati fra loro al fine del raggiungimento dell'obiettivo didattico generale del Corso.

Il nostro metodo di generazione di un generico corso, è dinamico e può essere così descritto:

Conosciuto lo stato iniziale **SIK** (Starting Inizial Knowledge) del discente, che descrive il livello di preparazione iniziale in termini di conoscenza che egli ha della materia trattata nel corso, conosciuto l'obiettivo formativo **TCK** (Target Course Knowledge) che il corso persegue, il software contenuto nell'LCMS individua una sequenza di Learning Objects (il cosiddetto *Path Learning*), che conduce il discente, passo dopo passo, in uno stato desiderato detto **LAKi** (Learner Actual Knowledge) (portino il discente in stati in cui il

⁵⁵ ELLADE, Elctronic Live ADptive Learning, progetto Univerisità della Calabria , documento reperito al link: <http://www.idtech.it/docs/ellade.pdf>,

gap fra la sua conoscenza attuale e quella richiesta è minima, fino ad arrivare in **TCK**).

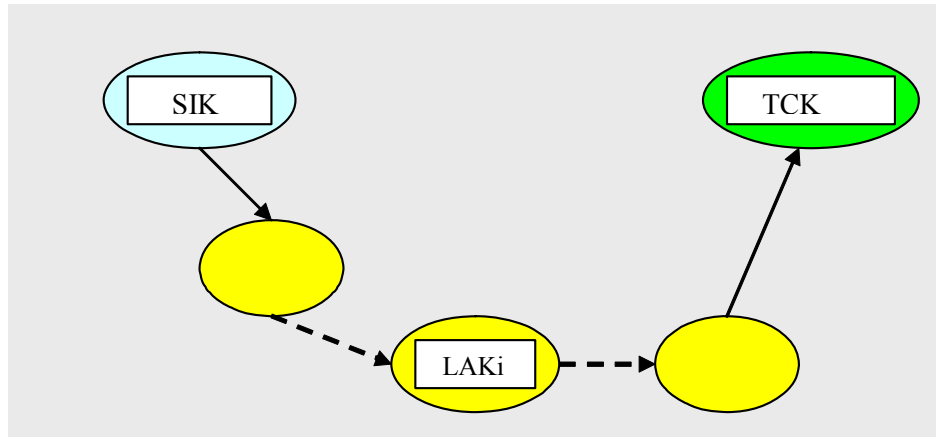


Fig. 5.1 - (Path Learning)

E' cosa abbastanza impegnativa individuare da parte degli LCMS un Path Learning ottimale poiché esso, sia non può essere individuato una volta per tutte, sia perché dipende da innumerevoli fattori, non ultima la quantità di materiale a disposizione su un determinato argomento appartenente al Path Learning in questione.

5.2 Tipologie di livelli di apprendimento e path learning statico

Il discente quando entra nella piattaforma LCMS per partecipare ad un certo corso esegue l'iscrizione. Affronta come primo passo prioritario un questionario (*test introduttivo*) che ha il compito di verificare quale preparazione (degli argomenti trattati al corso cui si è iscritto), possieda (preparazione iniziale). In base agli esiti, in termini numerici, riportati in tale prova, il learner seguirà “sarà dirottato verso”, uno stato corrispondente. Questo semplice meccanismo può essere descritto attraverso una tecnica che preveda: un Ingresso **I** (Risultato del test) al DeMultiplexer⁵⁶ che, lo smista verso lo Stato SIK0, SIK1, SIK2 o SIK3 (in base ai valori assunti da due variabili di controllo del DeMux S1 S0, due ingressi che con il loro valore binario identificano l'esito del test introduttivo). Lo stato SIK0 corrisponde a preparazione iniziale adatta ad un certo percorso propedeutico, lo stato SIK1 corrisponde a preparazione iniziale adatta al percorso principiante, lo stato SIK2 corrisponde a preparazione iniziale adatta al percorso intermedio, lo stato SIK3 corrisponde a preparazione iniziale adatta al percorso avanzato. S0 ed S1 sono due variabili binarie che possono assumere i valori 00, 01, 10 e 11.

Ad esempio se, dopo essersi cimentato con lo svolgimento del test introduttivo il learner, riporta un voto inferiore uguale a 3/30, è assegnato al percorso propedeutico (S1S0 valgono 00), se riporta un voto compreso fra 4/30 e 8/30 (S1S0 valgono 01), è assegnato al sottocorso per principianti, se riporta un voto compreso fra 9/30 e 15/30 (S1S0 valgono 10), è assegnato al percorso intermedio, per voti superiori a 15/30 è assegnato al percorso avanzato (S1S0 valgono 11).

⁵⁶ Sistemi Elettronici Digitali, Piero Marietti, G. Ciccarella, A. Londei, F. Loriga p. 139 ed. Masson, pag.90

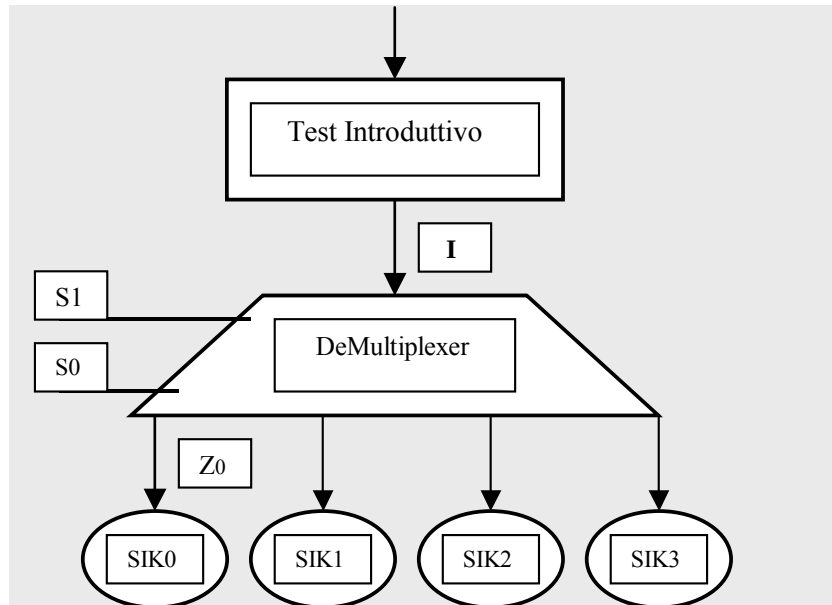


Fig. 5.2 - (Schema di selezione tipologia corso)

Se chiamiamo con Z_0, Z_1, Z_2, Z_3 le uscite del DeMultiplexer esse valgono:

$$Z_0 = I \bullet \bar{S}_1 \bullet \bar{S}_0$$

$$Z_1 = I \bullet \bar{S}_1 \bullet S_0$$

$$Z_2 = I \bullet S_1 \bullet \bar{S}_0$$

$$Z_3 = I \bullet S_1 \bullet S_0$$

che ci conducono negli stati rispettivi SIK_i , ($i=0..3$), rappresentanti il punto di ingresso del corrispettivo percorso.

Il percorso⁵⁸ assegnato allo studente è un *path learning statico* fissato in fase di costruzione del corso al fine di diminuire, sin dall'inizio in maniera drastica, il gap fra la conoscenza posseduta dal learner (SIK) e la conoscenza richiesta (TCK). Pensiamo al percorso assegnato come costituito da un certo numero di M Moduli.

⁵⁷ Il puntino • indica il simbolo della funzione logica And

⁵⁸ Qui il percorso didattico è un insieme di Moduli sequenziali preassegnati.

5.3 Diagramma di flusso arricchito con caratteristiche adattative “costruzione corso”

In questo paragrafo illustreremo il diagramma di flusso mostrato in figura 5.3, particolareggiato al caso di un corso contenente M Moduli, in cui si abbia NL SottoModuli di tipo lezione, NT⁵⁹ SottoModuli di tipo test di autovalutazione sui contenuti della lezione studiata ed M SottoModuli di tipo Prove intermedie, (una Prova Intermedia per ogni Modulo).

A tal riguardo iniziamo ad indicare il SottoModulo contenente la generica

$$SMod_{c, j_c, m_c, j_c, l_c, j_c, m_c, j_c}$$

lezione con la l_c, j_c, m_c, j_c ed il sottomodulo generico, contenente il test di autovalutazione, che pensiamo sia adiacente e successivo

$$SMod_{c, j_c, m_c, j_c, (l_c, j_c, m_c, j_c + 1)}$$

alla lezione lo indichiamo con $(l_c, j_c, m_c, j_c + 1)$, che assolve due compiti fondamentali:

a) Assegna i *crediti formativi* di Modulo (secondo quanto illustrato nel paragrafo 1.3.3) attraverso i risultati riportati dal learner nelle Prove Intermedie

b) Guida il discente nel suo *Path Learning* sulla base di due parametri:

b1) **RisTl+1** risultato numerico riportato, nel SottoModulo di tipo test di autovalutazione Tl+1,

con l=0 ... NT-1 (il valore di NT, numero di sottomoduli di tipo test o di

tipo lezione presenti in un modulo coincidente con $(L_{c, j_c, m_c, j_c} - 1)/2$ ⁶⁰

essendo L_{c, j_c, m_c, j_c} il numero totale di SottoModuli di un Modulo

Mod_{c, j_c, m_c, j_c}),

⁵⁹ NT coincide con NL abbiamo un test di autovalutazione per ogni lezione

⁶⁰ Il -1 è dovuto al fatto che in ogni Modulo consideriamo presente una sola Prova Intermedia

nel cimentarsi con il test di autovalutazione l -esimo (In termini generali

$$Ris_{SModc, j_c, m_c, j_c, (l_c, j_c, m_c, j_c + 1)}$$

avremo al posto di **RisT** $l+1$ la

$b2)$ attraverso il risultato numerico di **ObMt** ($t=1...M$)

$Ob_{SModc, j_c, m_c, j_c}$ “Obiettivo didattico del Modulo”, riportato nel SottoModulo, di tipo Prova Intermedia, corrispondente. (In generale M

coincide con la $M_{c, j_c} - 2$)⁶¹. Notiamo, dal diagramma di flusso, che il nostro sistema è di tipo “Open”, in pratica quando si raggiunge la fine di un modulo, non vincoliamo i learners al superamento delle Prove Intermedie, per avere accesso al modulo successivo in ossequio alla didattica centrata sul discente. Il punto 2) è stato così previsto. Se il risultato, **RisT** $l+1$, numerico

$$SModc, j_c, m_c, j_c, (l_c, j_c, m_c, j_c + 1)$$

conseguito nel $SModc, j_c, m_c, j_c, (l_c, j_c, m_c, j_c + 1)$ è sufficiente si passa al SottoModulo successivo, altrimenti il SottoModulo di tipo lezione

$SModc, j_c, m_c, j_c, (l_c, j_c, m_c, j_c)$ è ricombinato con l'introduzione di L.O. di approfondimento su argomenti inerenti alla od alle risposte errate date dai learners nel test di autovalutazione.

Qui va scelto un numero finito di possibili ricombinazioni corrispondenti ognuno ad un nuovo stato.⁶² (creazione di stati dinamicamente in base al feedback del learner).

Conclusa l'autovalutazione dell'ultimo SottoModulo si passa ad eseguire la Prova Intermedia (Prova di conoscenza del modulo studiato).

Attraverso questa prova si possono acquisire i crediti formativi; se il risultato della prova intermedia del modulo è inferiore al minimo valore previsto (obiettivo di modulo che in genere coincide con l'obiettivo didattico del corso, 18 ad esempio) si potrebbe riassemblare l'intero modulo didattico.

⁶¹ Il -2 si giustifica poiché vi è una prova iniziale ed una prova finale in ogni corso.

⁶² Questo è un punto molto delicato, in quanto dalla scelta del numero di domande ND del test di autovalutazione, dipende la quantità del numero di stati possibili. Noi assumiamo, per il tipo di scelta fatta di ricombinazione dei sottomoduli, come ND=4, che comporta 15 possibili nuovi stati. Vedi paragrafo successivo per il calcolo dei possibili nuovi stati. Comunque, indipendentemente dal numero di stati il max numero di L.O. da sostituire coincide con ND, se abbiamo un unico sostituto per L.O. In genere si utilizza ND=10 nei test di autovalutazione.

Idem si procede con gli altri Moduli fino ad arrivare al Test finale, il superamento del quale dà diritto al Voto finale (Voto finale = Voto Esame - crediti formativi assegnati); se il Voto esame è inferiore all'obiettivo didattico del corso (24 ad esempio), il learner non è promosso perciò si azzerano i crediti e si deve ricominciare il tutto.

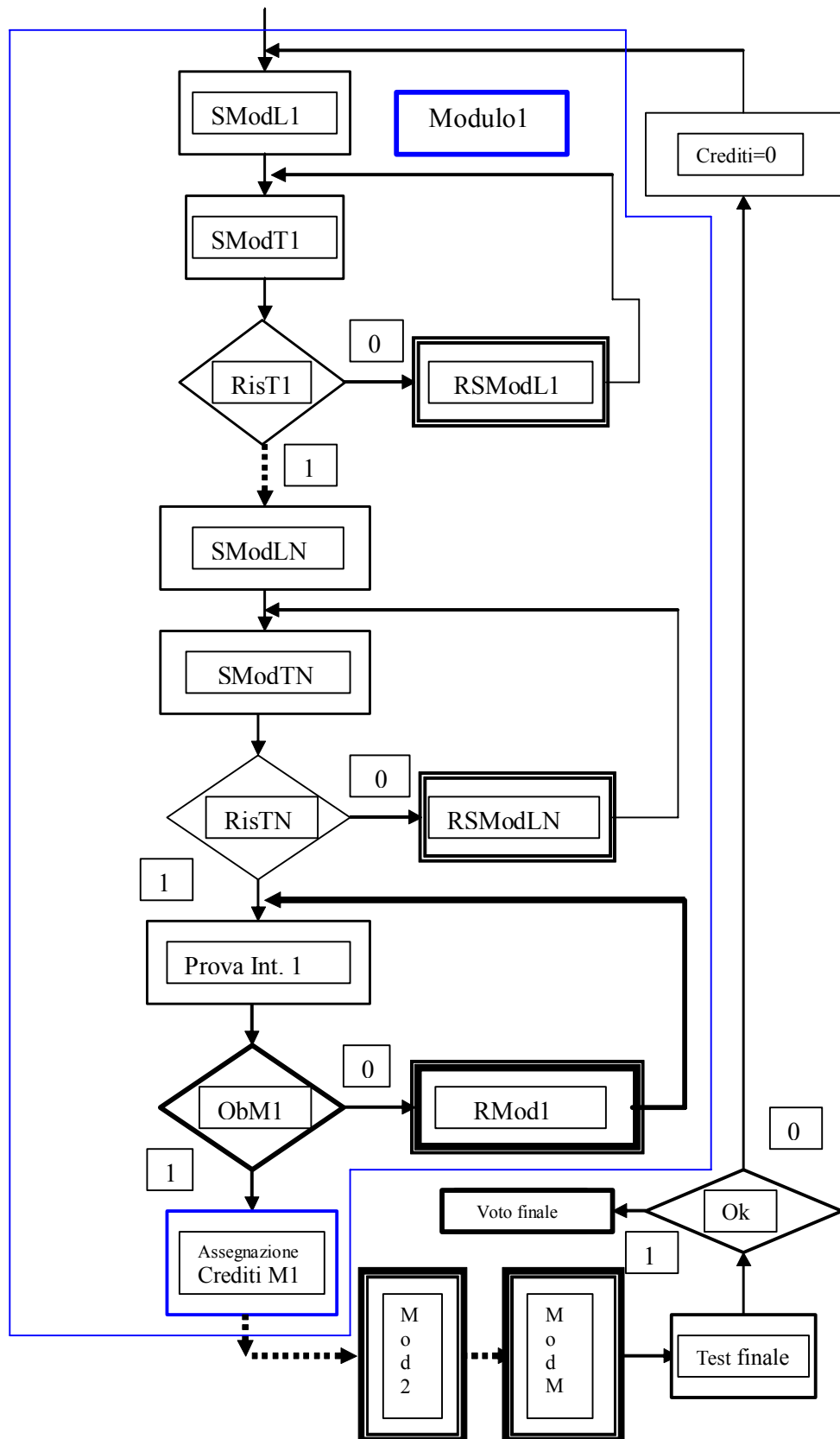


Fig. 5.3 - (Diagramma di flusso contenente il focus sull'apprendimento)

Nel nostro metodo ogni test di autovalutazione deve contenere almeno un Learning Object di tipo domanda/risposte per ogni L.O. appartenente al SottoModulo lezione corrispondente. Ossia il minimo numero di domande che deve contenere un test di di autovalutazione (Sottomodulo particolare “pseudo-atomico” poiché il suo contenuto fa dipendere il SottoModulo lezione) deve coincidere con il numero dei Learning Objects del SottoModulo Lezione con cui è stata aggregata, abbreviato $NDT \geq NLOL$. (Ricordiamoci che abbiamo indicato il SottoModulo contenente la generica lezione con la

$SMod_{c, j_c, m_c, j_c, l_c, m_c, j_c}$ ed il sottomodulo generico, contenente il test di autovalutazione, che pensiamo sia adiacente alla lezione lo indichiamo con $SMod_{c, j_c, m_c, j_c, l_c, j_c, m_c, j_c + 1}$).

Quando il learner esegue il test di autovalutazione le sue eventuali risposte errate le assumiamo essere una *misura* dei suoi punti di debolezza e su cui dovrà insistere nello studio.

Quindi con la scelta $NDT=NLOL$, sappiamo quali e quanti sono i Learning Objects da sostituire (nel caso non raggiunga l'obiettivo, di

$SMod_{c, j_c, m_c, j_c, l_c, j_c, m_c, j_c + 1}$, $RisTl+1$).

Possiamo quindi riassemblare il SottoModulo di tipo lezione percorrendo il ramo $RSMoDLl$ del nostro diagramma di flusso e ripresentarlo come nuovo stato ma medesima etichetta⁶³ SottoModulo $SModLl$.

Il learner riesegue il test di autovalutazione e si riautoverifica.

Lo pseudo-vincolo $RisTl+1$, nel diagramma di flusso alla fine del generico

$SMod_{c, j_c, m_c, j_c, l_c, j_c, m_c, j_c + 1}$ con $l = 0, 1, \dots ((L_{c, j_c, m_c, j_c} - 1)/2)$, che ci permette di far passare il learner allo studio della lezione successiva (altro nuovo stato) è dato dalla seguente formula (1.20):

$$Ris_{c, j_c, m_c, j_c, l_c, j_c, m_c, j_c + 1} = fatt_{30} * \sum_{v=1}^{NLOL} Voto_risp_esatte_v$$

$\geq Ob.Corsoc, j_c$

⁶³ Si aggiorna l'etichetta corrispondente al nuovo stato a quella del SottoModulo Lezione

(1.20)

Con **fatt30** abbiamo indicato il fattore di normalizzazione a 30 del voto riportato dal learner nel test di autovalutazione, mentre con **NDE** indichiamo il numero di domande cui ha risposto esattamente (risposte esatte), **Voto_risp_esatte** è il voto riportato nelle risposte corrette.

Indicando con **ND** il numero di domande contenute in un test di autovalutazione, il termine **fatt30** si ricava dalla formula seguente:

$$f_{att30c, j_c, m_c, l_c, m_c, l_c, m_c, l_c} + 1) = 30 / (\sum_{t=1}^{ND} peso_dc, j_c, m_c, l_c, m_c, l_c, m_c, l_c + 1) \cdot t)$$

(1.21)

Per le Prove intermedie, che consideriamo fisse e quindi non riassemblabili, si può procedere similmente per il calcolo del suo fattore di normalizzazione a 30, assegnando in generale un peso diverso alle diverse prove intermedie.

5.4 Esempio chiarificatore delle formule 1.20 e 1.21

Se abbiamo $ND = 4$ domande nel test di autovalutazione generico ed i pesi sono come in Tabella 5.1:

Tab. 5.1 - Calcolo fattore di normalizzazione

	D1	D2	D3	D4
peso d t	2	1	4	3
Voto risp.esattev	ok	no	ok	no

Applicando la 1.21

$$fatt_{30} = 30 / (2 + 1 + 4 + 3) = 30 / 10 = 3$$

mentre applicando la 1.20

$$Ris = 3 * (2 + 4) = 18 \text{ punti.}$$

Adesso in base alla 1.20 se 18 è maggiore od uguale al minimo obiettivo di SottoModulo, il learner può procedere allo studio del SottoModulo successivo con la consapevolezza di aver superato il Modulo attuale. Notare che lo pseudo-vincolo, per la psiche del learner, può essere il raggiungimento del massimo voto 30, superiore quindi all'obiettivo minimo fissato.

5.5 Calcolo numero stati del ramo di Riassemblaggio RSMOD

Calcoliamo adesso quante sono le possibili ricombinazioni⁶⁴ di SottoModuli di tipo lezione, che si generano ognuna per ogni errore commesso nel test di autovalutazione. Ipotizziamo che l'obiettivo del SottoModulo è rispondere esattamente a tutte le domande del test di autovalutazione. Nel caso in cui avessimo quattro domande il learner potrebbe commettere 0, 1, 2, 3 oppure 4 errori; il caso con 0 errori lo escludiamo poiché significa, come ovvio, che ha risposto correttamente a tutte le domande del test, ed in tal caso sicuramente, questo ramo, RSMOD del diagramma non è percorso⁶⁵.

Adottiamo nel nostro metodo i seguenti criteri:

⁶⁴ Ogni ricombinazione che si genera è un nuovo stato

⁶⁵ Dobbiamo prevedere che l'obiettivo del learner è quello massimo, raggiungere il 30 nel test di autovalutazione

a) Se ho un test di autovalutazione composto da n domande, e relative opzioni risposte, considero il test come formato da un aggregato di n^{66} Learning Objects.

Per ogni errore commesso nel test, si ricombina un SottoModulo di tipo lezione (sostituendo i Learning Objects corrispondenti agli errori commessi nel test, si cerca di trovare qualcosa maggiormente chiarificatore, puntando il focus sull'allievo e le sue capacità di apprendimento quindi).

b) Ipotizziamo di avere nella nostra biblioteca materiale a sufficienza per sostituire i Learning Objects citati nei punti precedenti; cioè ipotizziamo di avere un numero di nuovi L.O. pari al numero di domande del test di autovalutazione

La quantità totale di errori, $N_{tot.err}$, che si possono commettere in un test di autovalutazione composto da n domande è data dalla seguente formula:

$$N_{tot.err} = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n-1} + \binom{n}{n}$$

che in forma compatta può essere scritta come nella 1.22.

$$N_{tot.err} = \sum_{i=1}^n \binom{n}{i} = 2^n - 1 \quad (1.22)$$

Ad esempio per $n = 4$ domande avremo:

$$N_{tot.err} = \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4}$$

e ricordandoci la formula risolutiva dei coefficienti binomiali

$$\binom{n}{k} = \frac{n!}{k! * (n-k)!}$$

$$N_{tot.err} = 4 + 6 + 4 + 1 = 15$$

⁶⁶ n è quel che abbiamo chiamato ND numero di domande/risposte o Numero di L.O. del test di autovalutazione

Attraverso la formula 1.20 possiamo indicare quanti sono i possibili stati del blocco RSMODgenerico. Tenendo in considerazione il fatto che ogni errore mi origina un nuovo stato, il numero complessivo di stati generati coincide con il numero totale di possibili errori fatti dal learner nel test ed è pari a:

$$N_{tot.stati generati} = N_{tot.err} = \sum_{i=1}^n \binom{n}{i} = 2^n - 1$$

(1.23)

Con questo metodo si deve scegliere di costruire i test di autovalutazione con poche domande altrimenti avremo troppi stati generati e ciò complicherà di tanto la creazione di un percorso personalizzato.

5.6 Diagramma ASM per automatizzare il corso

In questo paragrafo, facendo uso dei diagrammi ASM Algorithmic State Machine, che è una metodologia di rappresentazione delle macchine a stati, da implementarsi in una piattaforma software, daremo degli spunti di riflessione per la progettazione di un percorso didattico personalizzato alle esigenze dei learners.

Gli elementi costituenti un diagramma ASM sono tre: le **caselle di stato**, le **caselle di decisione** e le **caselle di uscita**.

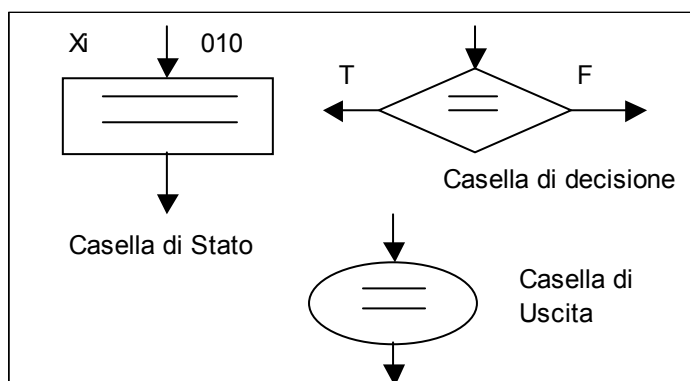


Fig. 5.4 - (Caselle ASM)

Ogni diagramma ASM è costituito da diversi blocchi, ognuno dei quali contiene una sola casella di stato (uno stato per ogni blocco) ⁶⁷.

Il problema che cercheremo di risolvere è quello della costruzione di un percorso didattico personalizzato per il learner; a tal fine consideriamo che il discente faccia parte di uno stato **SIK_i** (Starting Inizial Knwoledge) con ($i = 0,1,2,3$).

Detti $X(t)$ il vettore di stato, $U(t)$ il vettore degli ingressi ed $Y(t)$ il vettore delle uscite il modello di Mealy prevede le seguenti formule:

$$\begin{cases} X(t+1) = F_X(U(t), X(t)) \\ Y(t) = F_Y(U(t), X(t)) \end{cases} \quad (1.24)$$

⁶⁷ Sistemi elettronici digitali, Piero Marietti ed altri opera cit.

ed il modello di Moore:

$$\begin{cases} X(t+1) = F_X(U(t), X(t)) \\ Y(t) = F_Y(X(t)) \end{cases} \quad (1.25)$$

Introduciamo adesso una rete combinatoria chiamata **Multiplexer** o MUX che permette di selezionare uno fra diversi ingressi da presentare in uscita.

Una rappresentazione del Multiplexer⁶⁸ è mostrata in figura 5.5 per 4 ingressi I0, I1, I2, I3; in esso soltanto uno sarà presentato all'uscita Z. Agli ingressi S1, S0 si manda un codice in grado di selezionare quale degli ingressi Ii sarà presentato in uscita. Il numero delle linee di ingresso (Ii, Sj) è tale che se n_I è il numero degli ingressi **I** ed n_S è il numero degli ingressi **S** si abbia,

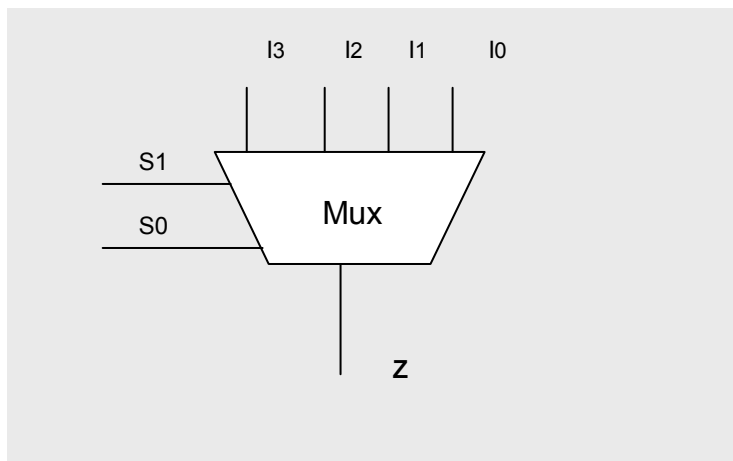
$$n_I = 2^{n_S}$$


Fig. 5.5 - (Schema di un Multiplexer)

La funzione logica che descrive il comportamento del Mux in figura 5.5 è:

$$Z = I_0 \cdot \overline{S_1} \cdot \overline{S_0} + I_1 \cdot \overline{S_1} \cdot S_0 + I_2 \cdot S_1 \cdot \overline{S_0} + I_3 \cdot S_1 \cdot S_0 \quad (1.26)$$

⁶⁸ Sistemi elettronici digitali, Piero Marietti et al. opera cit., p. 86

Per quanto detto nei precedenti paragrafi si ha che il ramo del grafo di flusso RSM_{l+1} con $l = 0, 1, \dots, ((L_{c, j_c, m_{c, j_c}} - 1)/2)$ riassume il sottomodulo corrispondente alla lezione $l+1$ -esima

Cioè in generale $RSM_{l, j_c, m_{c, j_c}, l_{c, j_c, m_{c, j_c}}}$ con $l = 1, 3, \dots, [L_{c, j_c, m_{c, j_c}} - 2]$

Organizziamo le cose in modo tale che se il learner risponde correttamente ad una domanda del Test di autovalutazione sia posta a 0 una corrispondente variabile binaria e se risponde in modo errato sia settata ad 1.

Nel nostro Test di autovalutazione utilizzato sinora come esempio abbiamo $ND=4$ per cui si avrà $n_i = 2^4 = 16$ ingressi I_i per $i=0 \dots 15$ e 4 ingressi, S_j per $j=0 \dots 3$, di controllo; cioè i possibili esiti del test di autovalutazione possono essere 16. Se traduciamo in binario ed indichiamo con **0 risposta esatta** e con **1 risposta errata** avremo la configurazione riportata in Tabella 5.2.

Tab. 5.2 - (Configurazione da esito del test di autovalutazione)

	D4	D3	D2	D1
I 0	0	0	0	0
I 1	0	0	0	1
I 2	0	0	1	0
I 3	0	0	1	1
I 4	0	1	0	0
I 5	0	1	0	1
I 6	0	1	1	0
I 7	0	1	1	1
I 8	1	0	0	0
I 9	1	0	0	1
I10	1	0	1	0
I11	1	0	1	1
I12	1	1	0	0
I13	1	1	0	1
I14	1	1	1	0
I15	1	1	1	1

Ad esempio la riga con ingresso I 5 esplicitata da **0 1 0 1** indica risposta alla domanda 4 esatta, risposta alla domanda 3 errata, risposta alla domanda 2 esatta, risposta alla domanda 1 errata; se tale ingresso è selezionato come uscita del Mux s'intende che va riassetata la lezione in questione sostituendo i L.O. di tipo lezione (con altri) corrispondenti ai L.O. (domanda 1)/risposte e (domanda 3)/risposte di tipo test di autovalutazione.

Riportiamo in Figura 5.6 il diagramma ASM parziale con cui il corso1,1 potrebbe essere riprogettato onde tener conto delle caratteristiche di apprendimento del learner. Ossia adattiamo il corso con eventuali nuovi SottoModuli o Nuovi Moduli in base al responso in termini di obiettivo didattico dato dal learner (*focus puntato sulle soluzioni di apprendimento*).

In questo modo sicuramente si riduce nelle varie fasi del corso il gap fra le conoscenze attuali del learner e le conoscenze obiettivo.

Nel caso si pensi di *riassetare soltanto i SottoModuli* di tipo *lezione*, e non l'intero Modulo, in caso di esito negativo dell'obiettivo di Modulo, avremo un numero massimo di possibili stati nel caso peggiore pari al [(numero di Moduli del corso -2)* il numero di lezioni contenute in un Modulo) * (il numero di combinazioni possibili L.O. per ciascuna lezione)]. In formula avremo cioè:

$$(M_{c \ j_c} - 2) * \left\{ \left[(L_{c \ j_c \ m_{c \ j_c}} - 1) / 2 \right] * 2^{ND} \right\} \quad (1.27)$$

In sostanza nella figura si applica il modello di Mealy per riconoscere una sequenza di 1 pari al numero di Moduli del corso per il numero di lezioni di

ogni Modulo, cioè $M * NL = (M_{c \ j_c} - 2) \left[(L_{c \ j_c \ m_{c \ j_c}} - 1) / 2 \right]$ (sequenza di $M * NL$ uni). Il **vettore di stato X** oltre che funzione del tempo è funzione del risultato del test di autovalutazione $RisTl+1$ che è un ingresso indicato con U nel modello Mealy.

$$X' = F_X(X, RisTl+1) \quad ^{69}$$

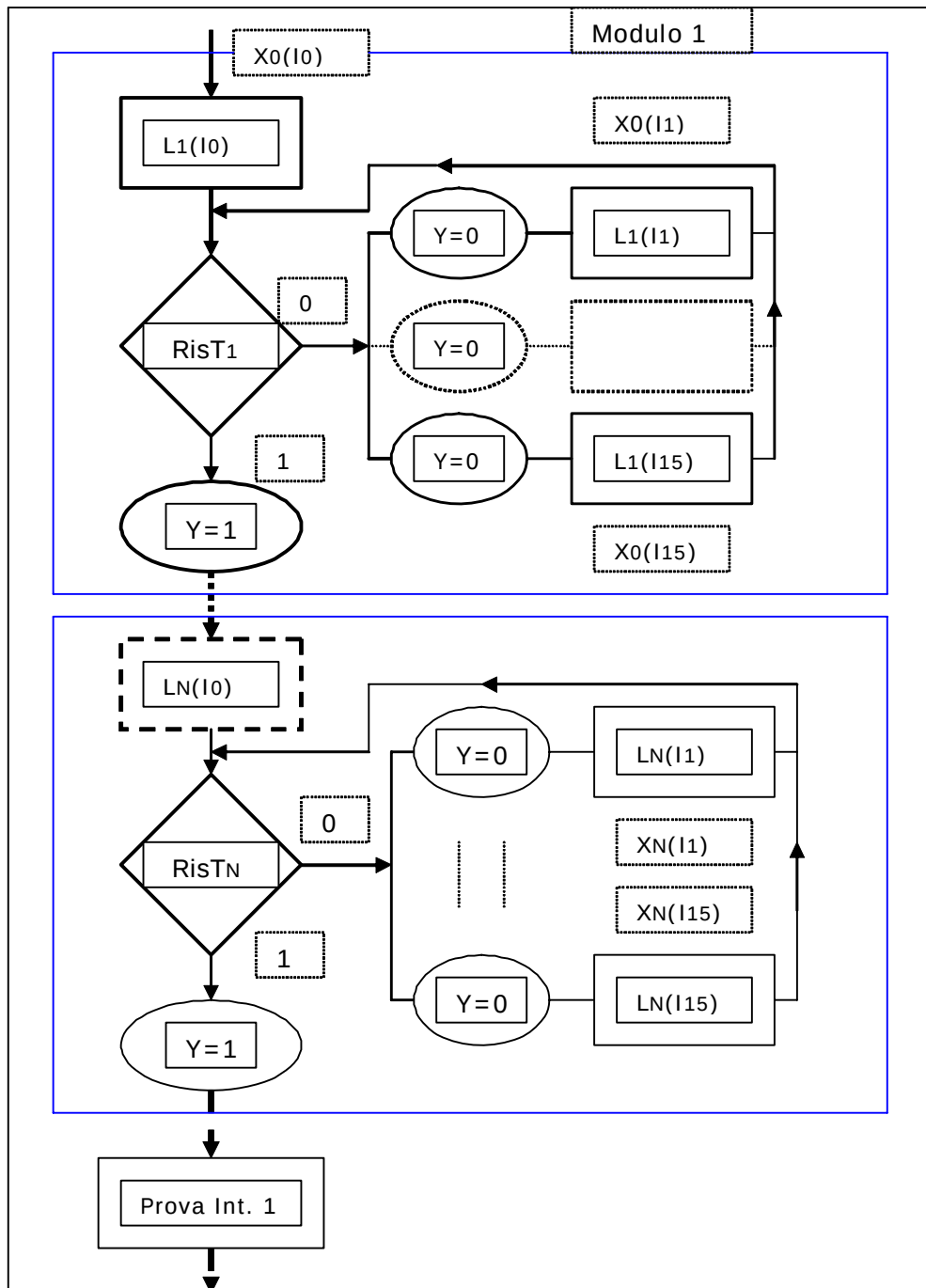
$$Y = F_Y(X, RisTl+1)$$

Dove $X = X(t, I_i)$; ossia lo stato attuale X è funzione sia del tempo sia del risultato del test di autovalutazione.

Nella pagina seguente, in fig. 5.6, mostriamo il diagramma ASM, parziale in termini di visibilità ma completo in quanto ripetitivo.

Fig. 5.6 - (Diagramma ASM parziale di Mealy adattativo)

⁶⁹ X' indica lo stato successivo, X indica lo stato attuale



6 Ruolo delle ontologie nei corsi e-Learning

6.1 Introduzione

La nostra metodologia di personalizzazione dell'apprendimento, sulle esigenze/preferenze, del discente abbiamo visto, nel capitolo 5, essere basata sul metodo della correzione di errore.

In particolare, abbiamo considerato come i SottoModuli di tipo Lezione si formassero aggregando fra loro learning objects corrispondenti didatticamente ai learning objects formanti i Test di autovalutazione.

Nel caso di errore, commesso dai discenti, nei test di autovalutazione nel nostro modello sono soltanto i L.O. di tipo Lezione ad essere sostituiti, prelevandoli da un **repository**⁷⁰ contenente L.O. di tipo lezione, per formare un nuovo SottoModulo (nuovo stato) che differisce, dal SottoModulo predecessore, soltanto nei L.O. sostituiti.

Abbiamo appreso *quanti L.O. sono sostituiti*, notando che nel caso peggiore occorre sostituire tutti i L.O. di un SottoModulo di tipo Lezione (che numericamente coincide con il numero di domande del corrispondente test di autovalutazione pari a ND); si è anche visto quale etichetta posseggono i L.O. da sostituire.

Rimane insoluto il problema di *come fare ad identificare il L.O. di tipo Lezione*, in genere contenuto nei **LOR** (Learning Objects Repositories), *potenziale sostituto*.

Purtroppo i metadata forniscono soltanto informazioni sui learning objects ma non dicono niente circa le relazioni intercorrenti fra di essi; precisamente non forniscono indicazioni sui *concetti* coinvolti nei learning objects e quindi

⁷⁰ Esistono già nel web molti “repository” come ad esempio MERLOT (Multimedia Educational Resource for Learning and Online Teaching) all'indirizzo <http://www.merlot.org>.

diventa complicato identificare un L.O., da presentare al discente o che il discente sceglie, se non si ha a disposizione una qualche relazione fra concetti, ossia qualcosa che espliciti il significato semantico dei vari L.O.

Una possibile soluzione a tal problema è quindi quella di introdurre una *ontologia* di dominio essendo lo scopo principale di una ontologia quella di permettere la condivisione ed il riutilizzo della conoscenza, prevedendo essa un meccanismo di relazione fra i contenuti di un L.O..

Altro problema cui bisognerebbe dare soluzione è quello della coincidenza del contenuto di un L.O. che può essere usato o contenuto in SottoModuli, Moduli, Corsi, Cataloghi diversi ed è etichettato con etichette diverse nel nostro modello.

Rispondere a quest'ultimo quesito è anche compito della ontologia di dominio che s'introduce per personalizzare l'apprendimento, magari associando all'etichetta dei L.O. uno stesso **URI** (Uniform Resource Identifier) identificatore univoco della risorsa, come avviene per le risorse *annotate*⁷¹ nel Semantic Web.

Un modo per risolvere quest'ultimo nostro quesito è quindi quello di *associare un medesimo URI alle diverse etichette dei L.O. che hanno lo stesso contenuto didattico*; difatti questi learning objects, generalmente sono collocati (od usati) nei vari SottoModuli i quali sono collocati (od usati) nei diversi livelli dello schema a livelli della biblioteca B.

Notare che nella visione costruttivista l'apprendimento è inteso come “costruzione” della propria conoscenza la quale si arricchisce e si ristrutturata nella negoziazione sociale dei significati.

Serena Alvino a tal proposito conclude che «I presupposti teorici su cui si basa il concetto di LO contrastano per alcuni aspetti con la visione dell'apprendimento del Costruttivismo Sociale.

Tuttavia, cercando di integrare queste due prospettive possono aprirsi nuovi orizzonti nella progettazione di unità didattiche a distanza:

- l'estensione del concetto di LO ad
 - esperienze, casi reali e prodotti di una comunità di apprendimento,
 - meta-conoscenze didattico-organizzative sviluppate dai progettisti di interventi formativi (design-time LO)
- l'inquadramento dei LO nell'ambito delle risorse riusabili e condivisibili nell'ambito di una Comunità di Pratica;
- una concezione dinamica ed evolutiva dei repository e degli stessi LO ⁷²»

⁷¹ L'*annotazione* è una delle tecniche più utilizzate per creare metadata sul Web oppure un metodo per inserire od associare metadati ad una risorsa.

⁷² *Learning Objects e Costruttivismo*, S. Alvino, L. Sarti, Didamatica 2004, Ferrara, (12/05/2004)

6.2 Ontologie

L'e-Learning consente la personalizzazione del percorso didattico, sulle esigenze/preferenze dei singoli discenti, se si utilizzano strumenti software che consentono di rendere dinamico il corso e quindi modificabile sia dai docenti, sia dal tutor e sia dai discenti in linea con la teoria costruttivista.

Vi sono delle problematiche che emergono nel far ciò e per risolverle necessita l'utilizzo di ontologie nella costruzione e fruizione di corsi in modalità e-Learning.

E' d'interesse, quindi, comprendere il concetto di ontologia; il termine ontologia è stato inizialmente utilizzato nelle scienze filosofiche in cui assumeva il significato di "*studio dell'essere in quanto tale*"; oggi tale termine è stato riadattato in altre discipline come l'Intelligenza Artificiale.

Esistono numerose definizioni di ontologia e sicuramente una definizione fra le maggiormente citate in letteratura è quella data da *Tom Gruber* nel 1993.

Secondo **Gruber** un'ontologia è una esplicita specificazione di una concettualizzazione.

Una concettualizzazione è l'insieme di oggetti, concetti ed altre entità che si può assumere esistere in una certa area di interesse e delle relazioni che esistono tra essi.

Gruber ha anche definito una ontologia come una quintupla composta di classi, istanze, funzioni, relazioni ed assiomi.

Le *classi* corrispondono alle entità del dominio, le *istanze* sono gli oggetti contenuti nel dominio, *funzioni* e *relazioni* collegano le entità al dominio; gli *assiomi* circoscrivono il senso e l'utilizzo delle classi, delle istanze, delle funzioni e delle relazioni.

Secondo **Borst** una ontologia è una formale specificazione di una concettualizzazione condivisa.

Secondo **Guarino** una ontologia è un insieme di assiomi logici progettati per considerare il senso di un vocabolario.

Una ontologia rappresenta una concettualizzazione condivisa di un certo dominio; essa contiene l'insieme dei concetti (entità, processi, attributi et. altro), le definizioni e le relazioni fra concetti.

Nei corsi in modalità e-Learning contenenti **RLO**⁷³ si esprime quindi il loro contenuto semantico attraverso concetti appartenenti ad una ontologia che descriva il dominio del corso.

⁷³ **Reusable Learning Object**; rappresenta una unità di apprendimento erogabile a distanza, portabile, accessibile e riutilizzabile, in quanto realizzata in conformità con gli standard internazionali di riferimento [Vercelli 2003]

Costruita l'ontologia su di essa si basa la ricerca semantica dei learning objects.

Possiamo pensare l'ontologia come la mappa dei percorsi che portano da un L.O. ad un altro od anche che tenga traccia dei percorsi didattici di un utente.

Ad esempio la piattaforma IWT (Intelligent Web Teacher), citata nel primo capitolo, utilizza, nel modello della conoscenza, anch'essa ontologie; in IWT le ontologie sono strutture a grafo che consentono di descrivere formalmente un dominio didattico attraverso la specificazione di un vocabolario di concetti e l'identificazione delle relazioni intercorrenti fra essi.⁷⁴

Rispettano gli standard **SHOE** (Simple HTML Ontology Extension, University of Maryland) e **DAML+OIL** (DARPA Agent Markup Language + Ontology Interchange Language)⁷⁵ e supportano le seguenti relazioni: **B** (Belongs to), **R** (Required) e **SO** (Suggested Order). **B** utilizzata per implementare una gerarchia di concetti. **cBd** significa che il concetto *c* è parte del concetto *d*.

R utilizzata per implementare la propedeuticità. **cRd** significa che *d* è un concetto pre-requisito per apprendere il concetto *c*.

SO utilizzata per implementare una propedeuticità più soft. **cSOd** significa che se è necessario apprendere sia *c* che *d* è conveniente apprendere prima *c* e poi *d*.

Con la relazione implicita **E** (Explained by), mantenuta nei metadati, è possibile infine collegare ciascun concetto ai learning objects che spiegano tale concetto. **cEl** significa che il concetto *c* è spiegato nel learning object *l*.

In IWT è il docente a creare l'ontologia ossia una mappa concettuale per rappresentare il flusso dei concetti obiettivo di un corso.

Costruire ontologie per un dominio di conoscenza non è problema banale; oggi esistono dei tools che aiutano nella modellazione/costruzione fornendo, fra le altre cose, anche una visualizzazione grafica dell'ontologia.

Un tool o ambiente di supporto di tal tipo per la modellazione di schemi ontologici è Protégè 2000 molto diffuso ed affidabile.

⁷⁴Nicola Capuano, *IWT: Una Piattaforma Innovativa per la didattica Intelligente su Web*
http://www.capuano.biz/journal_papers.asp

⁷⁵ DAML - progetto DARPA (Defence Advanced Research Project Agency) OIL (Ontology Interchange Language) - progetto europeo IST (OntoKnowledge) DAML+OIL - W3C. DAML+OIL nasce dall'unione delle due iniziative. Estensione del "DL subset" di RDF (Resource Description Framework). DAML+OIL sottomesso al W3C come base per la standardizzazione, creato il gruppo Web-Ontology (WebOnt). WebOnt crea il linguaggio OWL (Ontology Web Language) basato su DAML+OIL unione di DAML-ONT e OIL.

6.3 Cenni sul semantic web

Ogni giorno milioni di persone nel mondo accedono ai siti web per cercare informazioni. Per la presenza, nei server della rete delle reti (Internet), di miliardi di pagine web indicizzate attraverso gli spider dei motori di ricerca, sta diventando complicato ottenere in breve tempo e senza ambiguità le informazioni cercate.

Il web attuale basato sull'HTML pone notevoli problemi di "rumore", quando si devono cercare informazioni nel web attraverso i motori di ricerca, essendo notevole la quantità di pagine create e visitabili attraverso la rete Internet, ed ambiguo il significato di alcuni termini che utilizziamo per la ricerca.

Ad esempio se volessimo cercare il termine "Aida" inteso come l'aria del compositore Giuseppe Verdi, non vi è modo per i motori di ricerca, (così come oggi sono organizzate le pagine web e che in genere basano la ricerca su delle parole chiave da confrontare con quelle introdotte nelle pagine web dai progettisti), di effettuare tale ricerca in modo univoco.

In generale ci forniscono delle pagine non pertinenti (rumore) in quanto le macchine non comprendono se ci si riferisce con Aida ad un nome comune od all'opera del compositore Verdi o ad altro.

Con il web semantico ci si propone principalmente di superare tali problemi oltre che di avere dati sul web definiti e collegati in modo che possano essere utilizzati per l'automazione, l'integrazione ed il riutilizzo attraverso varie applicazioni.

Tim Berners-Lee il creatore del web semantico lo definisce come «un'estensione del web attuale in cui alle informazioni sono date un senso, un significato ben definito, migliorando in questo modo la cooperazione tra i computers e le persone (marzo 2001)».

Corrado Petrucco pensa al web semantico «come l'evoluzione di Internet da semplice sistema di recupero di documenti, ad un sistema "intelligente" in cui l'informazione sarà compresa da specifici software in grado di assistere l'utente a localizzarla e a rielaborarla in modo personalizzato e adattato alle proprie esigenze. L'e-Learning ed anche la didattica tradizionale dovranno necessariamente tenerne conto, proprio per i vantaggi che questa flessibilità può fornire al processo d'apprendimento».⁷⁶

Il web semantico deve avere tre livelli fondamentali; al livello più basso vi sono i *dati* ("learning objects"), poi vi sono i *metadati* che riportano questi

⁷⁶ Petrucco, C. , Le Prospettive Didattiche del Semantic Web, Atti Didamatica 2003.

dati ai concetti di uno *schema* (schema che è l'ontologia), nell'ontologia si esprimono le relazioni fra i concetti, che diventano classi di dati.

Per integrare la semantica nel web si utilizzano principalmente le **annotazioni**.

Esse rappresentano un sistema per descrivere il contenuto di una risorsa web, rappresentando un metodo per inserire od associare metadati ad una risorsa.

Le annotazioni sono in grado di stabilire delle relazioni tra un oggetto esistente e la classe di una ontologia, attraverso la dichiarazione di triple (oggetto, concetto, attributi) in linguaggi basati sullo XML (RDF: Resource Description Framework, RDFschema, DAML+OIL, OWL: Web Ontology Language).

Prima di trattare brevemente dei linguaggi del semantic web, rileviamo che un URI serve ad identificare gli oggetti e/o risorse mentre la **URL** (Uniform Resource Locator) *identifica e localizza* una risorsa.

In generale un URI non dà istruzioni al computer su come trovare una risorsa dà ad esso maggiori informazioni su una risorsa.

Nella URI **RFC1630** seguente notiamo come in essa è inglobata la URL. Quindi un **URI** = **URN** + **URL** (dove l'URN è Uniforme Resource Names)

[RFC1630]

"Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", T. Berners-Lee, Giugno 1994.

Disponibile in <http://www.ietf.org/rfc/rfc1630.txt>.

Ad esempio con <http://www.comunedasa.it/mimc/index2.asp> si indica un URL che identifica e localizza la risorsa "home page del mio sito web".

Il linguaggio XML, la cui prima bozza è del 1996, è un linguaggio di Markup pensato per scambiare in modo semplice documenti via web; consente a chiunque di progettare il proprio formato dei documenti e di scrivere poi in quel formato.

Le componenti di XML sono: il *contenuto* (XML); le specifiche che riguardano gli elementi ossia la *struttura* (DTD, Document Type Definition o XMLSchema) e le specifiche che riguardano la visualizzazione ossia lo *stile* (XSL eXtensible Style Language).

Vediamo a titolo esplicativo un esempio di contenuto XML:

```

<?xml version= "1.0"
<biblioteca>
  <libro>
    <titolo>titolo del libro</titolo>
    <autore>nome dell'autore</autore>
    <data>data pubblicazione</data>
    <ISBN>codice libro</ISBN>
    <editore>nome editore</editore>
  </libro>
</biblioteca>

```

Un documento o contenuto XML è un insieme correttamente annidato di tag aperti e chiusi (elementi): <titolo>titolo del libro</titolo>.

In XML gli elementi possono avere un numero qualunque di coppie *attributo-valore*: <titolo carattere= "incisione argento">Ricordanze della mia vita</titolo>.

Un documento XML che sia sintatticamente corretto è detto *ben formato*. In XML il *namespace* è una collezione di nomi identificati da URI.

Il W3C (World Wide Web Consortium) attraverso il gruppo Web-Ontology (WebOnt) con una Recommendation del 14 febbraio 2004 crea il linguaggio OWL (Web Ontology Language) basato sui linguaggi DAML+OIL.

OWL è il linguaggio su cui sta convergendo la comunità scientifica del settore per descrivere le ontologie in quanto si propone adatto a superare i problemi dei precedenti linguaggi.

7 Appendice A -Test finale corso

7.1 Test finale del corso di FI-ODL

"Test finale per gli studenti di fondamenti di informatica II "

Questo test serve per valutare il livello finale della preparazione raggiunta dai partecipanti al corso, al fine di analizzare se il corso è stato utile.

Buon lavoro.

Compilate il questionario in tutte le sue parti ed inviate

Cognome

Capano

Nome

Domenico

Num. di matricola :

F.C.

E-Mail:

domenico.capano@fastwebnet.it

Corso :

Fondamenti 2

Leggete attentamente le domande del test e date la/le vostra/e risposta/e .

1) Durante quale fase del processo di compilazione viene iniziata la costruzione della tavola dei simboli?

- ☐ a) Caricamento
- ☐ b) Analisi sintattica
- ☒ c) Analisi lessicale
- ☐ d) Collegamento

2) Quale e' il significato della istruzione `if (feof(stream));` essendo il prototipo della funzione `feof: int *feof(FILE *stream);`

- O a) La istruzione verifica se si e' raggiunto l'indicatore di fine file; essa ritorna il valore 1 se il test e' vero (ossia se si e' raggiunto EOF) oppure un valore uguale ad 0 se il file e' finito.
- O b) La istruzione verifica se l'indicatore di file si e' posizionato all'inizio del file; essa ritorna il valore 0 se il test e' falso (ossia se non si e' posizionato all'inizio del file) oppure un valore diverso da 0 se il posizionamento e' avvenuto in modo corretto.
- O c) La istruzione verifica se l'indicatore di file si e' posizionato all'inizio del file; essa ritorna il valore 1 se il test e' vero (ossia si e' posizionato all'inizio del file) oppure un valore uguale a 0 se il posizionamento non e' avvenuto in modo corretto.
- d) La istruzione verifica se si e' raggiunto l'indicatore di fine file; essa ritorna il valore 0 se il test e' falso (ossia se non si e' raggiunto EOF) oppure un valore diverso da 0 se il file e' finito.

3) Qual è il fine della parola chiave `typedef` nel linguaggio C?

- a) Fornire un meccanismo per creare dei sinonimi o pseudonimi, per i tipi di dato definiti in precedenza.
- O b) Utilizzo solo e soltanto per evitare l'uso delle structure tag.
- O c) Creare nuovi tipi di dato che non esistono nel C standard.
- O d) Definire nuove variabili per il tipo `int`, `char` e `float`

4) Il seguente pseudocodice, supponendo $n > 0$:
`int a, b, n, somma=0; leggi a, b, n ; if(a!=b) allora`
`somma=(somma+1) * n *(n+1)/2); write(somma);`
 quale problema risolve ?

- O a) stampa il valore $[n*(n+1)/2 + 1] * [n*(n+1)/2]$ soltanto se a e' diverso da b .
- O b) stampa la somma dei numeri naturali da 1 fino ad n soltanto se a e' diverso da b .
- O c) non funziona in quanto nella dichiarazione alla variabile `somma` abbiamo assegnato il valore zero.
- O d) stampa il valore $[n*(n+1)/2 + 1]$ soltanto se a e' diverso da b .

5) La rappresentazione degli interi relativi X , nell'intervallo $[-(b^n)/2, (b^n)/2)$, tramite il residuo modulo b^n di X , e' detta:

- O a) Rappresentazione in base b degli interi
- O b) Rappresentazione Ascii dei numeri in base b

- c) Rappresentazione in complemento alla base b
- O d) Rappresentazione in modulo di un intero in base b

6) Cosa fa la funzione DO definita nel programma seguente:

```
#include<stdio.h>
#define TRUE 1
#define FALSE 0
typedef int booleano;
typedef int TipoElementoMatrice;
typedef TipoElementoMatrice TipoMatrice[4][4];
TipoMatrice matrice;
```

```
booleano DO(TipoMatrice matrice);
int main(void)
{
    /* ... */
    return 0;
}
```

```
booleano DO(TipoMatrice matrice)
{
    int i,j;
    for(i=0;i<4;i++)
    {
        for(j=0;j<4;j++)
        {
            if(i!=j&&matrice[i][j]!=0)
                return FALSE;
        }
    }
    return TRUE;
}
```



7) Quale problema risolve il seguente algoritmo , se r e' un numero intero maggiore di 1 ?

1) leggi r; 2) cont <--- 1; 3) cont = r * cont; 4) somma= 1;
5) se (cont <= r) {somma <--- somma + cont; cont <--- cont * 2;}
6) stampa somma;

- O a) Stampa della somma di tutti i numeri da 1 fino ad r+1
- O b) Stampa del valore r+1
- O c) Stampa, del valore $2r + 1$
- O d) Stampa, del valore $2r$

8) Quale/i fra le opzioni sono corrette nel linguaggio C, avendo la seguente dichiarazione di struttura (*selezionare tutte le opzioni di interesse*):

```
struct ind { char nome[30]; char indir[40]; char citta[20]; } ind_info;
```

- ☐ struct ind{char nome[30]; char indir[40]; char citta[20];}indirizzario;
- ☐ struct ind *pun;
- ☐ printf("%s\n",pun->nome);
- ☐ ind_info.indir[29]='c';

9) Avendo le seguenti:

```
#define MAX_NODI 100;
```

```
typedef enum {TRUE=1,FALSE=0} Booleano;
```

```
typedef int Tipo_atomi;
```

```
struct Struct_albero_bin {  
    Tipo_atomi info;  
    Booleano esiste;  
};
```

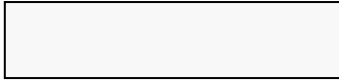
```
typedef struct Struct_albero_bin Tipo_nodo_albero;
```

```
typedef Tipo_nodo_albero Tipo_albero_bin[MAX_NODI];
```

Avendo la seguente funzione:

```
void analizza(Tipo_atomi info){printf("%d", info);}
```

scrivete una funzione ricorsiva in C, che esegue la visita in preordine dell'albero binario su rappresentato ; la funzione in questione riceve come parametri **albero** di tipo **Tipo_albero_bin**, in cui sono memorizzati i valori dei nodi e l'intero **indice**, che rappresenta la posizione dentro l'array **albero** in cui e' memorizzata la radice dell'albero, se esso non e' vuoto.



10) La seguente funzione C :

```
long int F (int k) {if(k==0) return 1; else return k*F(k-1); }
```

 cosa utilizza e per calcolare che cosa ?

- O a) usa la ricorsione per calcolare la somma dei numeri interi da 1 fino a k.
- O b) usa la ricerca binaria e calcola la somma dei numeri interi da 1 fino a k.
- c) usa la ricorsione per calcolare il fattoriale di k.
- O d) usa la ricorsione per calcolare il fattoriale di k, ma la funzione e' non corretta.

11) Avendo la seguente definizione

```
struct nodo {int info; struct nodo *prox;};
```

 cosa fa questa funzione:

```
void func(struct nodo *alb) {while (alb!=NULL)
```

```
    printf("%d--->", alb->info); alb=alb->prox; } printf("NULL\n"); }
```

- O a) Stampa l'albero binario partendo dall'elemento puntato dal puntatore alb ricevuto come parametro..
- O b) Stampa la lista partendo dall'elemento puntato dal puntatore alb ricevuto come parametro..
- O c) Stamperebbe la lista partendo dall'elemento puntato dal puntatore alb ricevuto come parametro se la condizione nel ciclo while fosse (prox!=NULL).
- O d) Stampa gli interi del campo info dell'albero binario il cui puntatore alla radice e' alb.

12) Eseguite ognuna delle seguenti attività:

Scrivete l'intestazione di una funzione chiamata **scambia** che riceva come parametri due puntatori ai numeri in virgola mobile n ed m e che non restituisca alcun valore.

```
void scambia(float *n, float *m)
```

b) Scrivete il prototipo per la funzione della parte a)

```
void scambia(float *, float *);
```


c) Scrivete l'intestazione di una funzione chiamata **evaluate** che ricevendo un intero n ed un puntatore alla funzione **poly** (*funzione intera con un argomento intero*), restituisca un intero ottenuto calcolando la funzione **poly** su n.

d) Scrivete il prototipo per la funzione della parte c)

13) Avendo la seguente definizione:

```
struct tabl
```

```
{ char nome[38];  
  char codice[5];  
  char telefon[20];  
  char fax[20];  
  char cap[6];  
  char citta[20];  
  char indiriz[45];  
  char note[51];  
}
```

CL1[100]; cosa rappresenta tale dichiarazione, se e' corretta ?

14) Tenendo in considerazione la domanda 13) il seguente codice:

```
int t=3; for(t=0;t<=99;t++) strcpy(CL1[t].nome,"nessuno");
```

15) Determinate quali fra le seguenti affermazioni sono vere e quali sono false: (*facoltativamente potete scrivere la motivazione della risposta*)

a) Se il puntatore di posizione di un file sequenziale fa riferimento ad un byte diverso dall'inizio di un file, quest'ultimo dovrà essere chiuso e riaperto per leggere nuovamente da quel punto (inizio file).

motivazione :

b) La funzione fseek, può ricercare solo dall'inizio di un file.

motivazione :

c) La funzione fprintf, può scrivere sullo standard di output.

motivazione :

d) Per trovare un certo record in un file ad accesso casuale, non e' necessario scorrere tutti gli altri record.

motivazione :

e) I record dei file ad accesso casuale non hanno una lunghezza uniforme.

motivazione :

16) Nel programma seguente cosa fa la funzione nome_funz?

```
#include<stdio.h>
#include<stdlib.h>
struct tab1
{ char carattere;
  struct tab1 *next;} info;

int main()
{
void nome_funz(char *);
char car;
char stringa[35];
printf("\ninserite una stringa\n");
gets(stringa);
nome_funz(stringa);
scanf("\n%c",&car);
return(0);
}
void nome_funz( char *strin )
{struct tab1 *iniz;
char car;
if(*strin!=NULL)
{
  iniz=malloc(sizeof(struct tab1));
  iniz->carattere=*strin;
  iniz=iniz->next;
  nome_funz(strin+1);
}
iniz=NULL;
}
```

- O a) costruisce ricorsivamente una pila, aggiungendo ogni nuovo carattere di strin all'inizio
- O b) cancella l'ultimo carattere di strin in una lista semplice
- O c) costruisce ricorsivamente una lista doppia, aggiungendovi ogni nuovo carattere di strin alla fine
- O d) costruisce ricorsivamente una lista semplice, i cui nodi contengono i caratteri di strin

17) Avendo la definizione

```
struct tab1
{ char nome1[38];
  char codice[5];
  char telefon[20];
  char fax[20];
  char cap[6];
  char citta[20];
  char indiriz[45];
  char note[51];
  struct tab1 *next;
  struct tab1 *precedente;
} lista_doppia;
```

E' possibile creare una lista doppia ordinata?

18) Come e' definito un albero binario di ricerca ?

Definiamo albero binario di ricerca un albero binario A, in cui ogni nodo x ha la proprietà: tutti i nodi del sottoalbero sinistro di x hanno un valore minore od uguale del valore del nodo x e tutti i nodi del sottoalbero destro hanno un valore maggiore di quello del nodo x.

19) Avendo definito, una struct albero, con il seguente prototipo di funzione:

```
struct albero *fun_alb( struct albero *radice, struct albero *r, char info);
```

e' possibile costruire un albero binario di ricerca usando la funzione fun_alb ?

- O a) Si ma fun_alb deve restituire un void
- O b) No
- O c) Si
- O d) Si ma non ordinato

20) Nella macchina virtuale del C nel modello a tempo di esecuzione, abbiamo le seguenti aree di memoria

- a) area del codice
- b) area dati globali

c) area heap

d) area stack

elencate quali sono le funzioni che svolgono queste aree di memoria?

21) Un array monodimensionale di nome `vector`, di 10 caratteri, come si può dichiarare nel linguaggio C?

```
char vector[11];
```

22) la seguente dichiarazione `typedef int k[8] tipoa;` cosa rappresenta?

23) $(11000.1011)_2$ corrisponde a : in base 10.

24) C2A4 esadecimale corrisponde a : in base 8.

25) Supponete di dover utilizzare una lista doppia in un programma C, definite la lista mediante strutture e puntatori.

26) Perché nel passaggio di un array ad una funzione, quest'ultimo viene sempre passato per indirizzo nel linguaggio C?

27) E' lecita la seguente definizione di struttura: `struct{int giorno;
int mese; int anno;}data;`

- ☐ NO non ci vuole il punto e virgola alla fine di **data**
☒ SI
☐ NO, manca il nome della struttura
☐ SI, poiché tutti i membri sono dello stesso tipo

28) Cosa fa la funzione `malloc`, il cui prototipo e':
`void *malloc(size_t dim);`

29) Il numero di nodi di un albero binario completo, avente profondità h e' pari

a : supponendo, che la radice abbia profondità zero.

30) Nella funzione selection-sort, per ordinare 10 elementi di tipo carattere contenuti in un array, quanti scambi sono necessari nel caso peggiore (tutti gli elementi sono fuori posto)

- ☐ a) 50
☐ b) 51
☐ c) 43
☒ d) 45

Tempo di inizio: 2:06:24 P.M. Tempo finale: 2:36:34
Tempo impiegato 30' 10''

Controlli la correttezza dei dati ed invii il modulo

N.B. Il modulo va inviato una sola volta

Invia Reimposta

8 Appendice B

8.1 Questionario di valutazione del Corso FIODL

In questo questionario le vostre risposte, servono a valutare il corso ed i suoi docenti.

Completarlo in tutte le sue parti.

Le vostre risposte ci saranno molto utili per capire se abbiamo fatto errori e come eventualmente correggerli in futuro.

Osservazioni e commenti, anche non direttamente suscitati dalle domande, saranno graditi.

1) Perché hai deciso di seguire il Corso FI-ODL ?

(selezionare tutte le opzioni di interesse)

- ☒ Mi interessava sperimentare la didattica a distanza
- ☐ Mi interessavano gli argomenti trattati nel corso
- ☐ Mi interessava avere materiale aggiuntivo sulla materia

Altri motivi

2) La tua motivazione prima di iniziare il corso era:

- ☐ Molto alta
- ☐ Alta
- ☐ Sufficiente
- ☐ Scarsa

3) La qualità del materiale didattico fornito nel corso è:

- ☐ Molto buona
- ☐ Buona
- ☐ Sufficiente
- ☐ Insufficiente
- ☐ Mediocre

4) Hai trovato utili per l'apprendimento i test di autovalutazione?

- ☐ Molto
- ☐ Sufficientemente
- ☐ Scarsamente

5) Ritieni che, in generale, ciascuna lezione sia stata:

- ☐ Troppo lunga
- ☒ Della giusta lunghezza
- ☐ Troppo corta

Eventuali commenti più specifici

6) Esempi e prove proposte sono, in numero e qualità:

- ☐ Molto buoni
- ☐ Buoni
- ☐ Sufficienti
- ☐ Insufficienti
- ☐ Mediocri

7) Le prove intermedie sono state:

- ☐ Difficili ma autoistruttive
- ☐ Difficili
- ☐ Di normale difficoltà
- ☐ Facili

8) Le informazioni ricevute dai tutors durante il corso sono state:

- ☐ Ottime
- ☐ Più che sufficienti
- ☐ Sufficienti
- ☐ Insufficienti

9) Ritieni che le informazioni e gli aggiornamenti siano pervenuti in tempi:

- ☐ Troppo veloci
- ☐ Veloci
- ☐ Sufficientemente veloci
- ☐ Scarsamente veloci

10) La frequenza con cui hai visitato le pagine del sito è stata:

☐ Più volte al giorno

☐ Una volta al giorno

☐ Una volta ogni due giorni

☐ Settimanale

☐ Altro

11) Durante il corso, la tua richiesta di informazioni è stata rivolta più frequentemente a:

☐ Docenti

☐ Colleghi di corso

☐ Nessuno

☐ Altri

12) Quanto sostegno hai trovato nei tuoi docenti?

☐ Più che sufficiente

☐ Sufficiente

☐ Scarso

13) Quanto sostegno hai trovato nei tuoi colleghi?

☐ Più che sufficiente

☐ Sufficiente

☐ Scarso

14) Durante il corso, hai avuto la sensazione di sentirti isolato?

☐ Sì, molto

☐ Sì, un pò

☐ No

15) Ritieni che sia stato facile adattarti a questo tipo di corso?

☐ Sì

☐ Abbastanza

☐ No

16) Il corso che stai valutando è stato tenuto via internet.

Considerando i contenuti e le attività svolte, ritieni che un tale tipo di corso sia: complessivamente migliore di uno analogo svolto in modo tradizionale?

☐ Sì

☐ No

☐ Non vedo differenze sostanziali

☐ Non so

17) Quali sono stati i vantaggi nel seguire un corso di questo tipo?

☐ Seguire le lezioni da casa

☐ Scegliere il momento in cui studiare le lezioni ed eseguire le attività connesse

☐ Non ci sono stati vantaggi

Altro

18) Ritieni che un corso a distanza possa sostituire uno tradizionale?

☐ Sempre

☒ In molti casi

☐ In pochi casi

☐ Mai

19) Se hai risposto "in molti casi" o "in pochi casi" alla domanda precedente, puoi indicarci quali corsi potrebbero essere, organizzati via internet e quali no?

20) Pensi che questo corso abbia arricchito le tue competenze?

☐ Sì

☐ No

☐ Non so

21) Rispetto alle tue aspettative, il corso è stato:

O Migliore

O Uguale

O Inferiore

22) Il tuo giudizio complessivo su questo corso è:

O Molto buono

O Buono

O Sufficiente

O Insufficiente

23) In futuro, seguiresti nuovamente un corso di didattica aperta a distanza?

O Sì

O No

O Non so

24) Dovendo essere tu a dare un voto complessivo al corso ed ai suoi docenti, quale sarebbe da 1 a 30 ?

25) Osservazioni e commenti:

Invia Reimposta

8.1.1 Risultati Valutazione del Corso FI-ODL

1) Perchè hai deciso di seguire il corso FI-ODL ? (selezionare tutte le opzioni di interesse)	
☐	Mi interessava sperimentare la didattica a distanza 50%
☐	Mi interessavano gli argomenti trattati nel corso 50%
☐	Mi interessava avere materiale aggiuntivo sulla materia 83%
	Altri motivi 50%
2) La tua motivazione prima di iniziare il corso era:	
☐	Molto alta 17%
☐	Alta 83%
☐	Sufficiente 0%
☐	Scarsa 0%
3) La qualità del materiale didattico fornito nel corso e':	
☐	Molto buona 17%
☐	Buona 83%
☐	Sufficiente 0%
☐	Insufficiente 0%
☐	Mediocre 0%
4) Hai trovato utili per l'apprendimento i test di autovalutazione?	
☐	Molto 67%
☐	Sufficientemente 17%
☐	Scarsamente 16%
5) Ritieni che, in generale, ciascuna lezione sia stata:	
☐	Troppo lunga 0%
☐	Della giusta lunghezza 100%
☐	Troppo corta 0%
Altri commenti più specifici 0%	
6) Esempi e prove proposte sono, in numero e qualità:	

☒	Molto buoni	50%
☒	Buoni	50%
☐	Sufficienti	0%
☐	Insufficienti	0%
☐	Mediocri	0%
7) Le prove intermedie sono state:		
☒	Difficili ma autoistruttive	33%
☐	Difficili	0%
☒	Di normale difficoltà	67%
☐	Facili	0%
8) Le informazioni ricevute dai tutors durante il corso sono state:		
☒	Ottime	50%
☐	Più che sufficienti	33%
☐	Sufficienti	17%
☐	Insufficienti	0%
9) Ritieni che le informazioni e gli aggiornamenti siano pervenuti in tempi:		
☐	Troppo veloci	0%
☒	Veloci	67%
☐	Sufficientemente veloci	33%
☐	Scarsamente veloci	0%
10) La frequenza con cui hai visitato le pagine del sito è stata:		
☒	Più volte al giorno	50%
☐	Una volta al giorno	0%
☐	Una volta ogni due giorni.....	17%
☐	Settimanale	33%
☐	Altro	0%

11) Durante il corso, la tua richiesta di informazioni è stata rivolta più frequentemente a:

☒	Docenti	67%
☒	Colleghi di corso	0%
☒	Nessuno	33%
☒	Altri	0%

12) Quanto sostegno hai trovato nei tuoi docenti?

☒	Più che sufficiente	50%
☒	Sufficiente	50%
☒	Scarso	0%

13) Quanto sostegno hai trovato nei tuoi colleghi?

☒	Più che sufficiente	17%
☒	Sufficiente	33%
☒	Scarso	50%

14) Durante il corso, hai avuto la sensazione di sentirti isolato?

☒	Sì, molto	17%
☒	Sì, un po'	16%
☒	No	67%

15) Ritieni che sia stato facile adattarti a questo tipo di corso?

☒	Sì	67%
☒	Abbastanza	33%
☒	No	0%

16) Il corso che stai valutando è stato tenuto via internet. Considerando i contenuti e le attività svolte, ritieni che un tale tipo di corso sia: complessivamente migliore di uno analogo svolto in modo tradizionale?

☐ Si 33%

☐ No 17%

☐ Non vedo differenze sostanziali..... 16%

☐ Non so 33%

17) Quali sono stati i vantaggi nel seguire un corso di questo tipo?

☐ Seguire le lezioni da casa 17%

☐ Scegliere il momento in cui studiare le lezioni ed eseguire le attività connesse 67%

☐ Non ci sono stati vantaggi 0%

Altro: 0%

18) Ritieni che un corso a distanza possa sostituire uno tradizionale?

☐ Sempre 33%

☐ In molti casi 67%

☐ In pochi casi 0%

☐ Mai 0%

19) Se hai risposto "in molti casi" o "in pochi casi" alla domanda precedente, puoi indicarci quali corsi potrebbero essere, organizzati via internet e quali no: 75% quasi tutti; 25% metodo tradizionale

20) Pensi che questo corso abbia arricchito le tue competenze?

☐ Si 100%

☐ No 0%

☐ Non so 0%

21) Rispetto alle tue aspettative, il corso è stato:

☐ Migliore 50%

☐ Uguale 50%

☐ Inferiore 0%

22) Il tuo giudizio complessivo su questo corso è:

☒	Molto buono	67%
☒	Buono	33%
☐	Sufficiente	0%
☐	Insufficiente	0%

23) In futuro, seguiresti nuovamente un corso di didattica aperta a distanza?

☒	Si	100%
☐	No	0%
☐	Non so	0%

24) Dovendo essere tu a dare un voto complessivo al corso ed ai suoi docenti, quale sarebbe da 1 a 30? .. 27

8.2 Un esempio di risposta al questionario di valutazione di FIODL pervenuta.

From: "Sab***" <***sab@oneonline.it>
To: capano.domenico@inwind.it, fondII.elt.*****@dis.uniroma1.it
Date: venerdì 16 novembre 2001 2.59
Subject: Valutazione del Corso FI-ODL

- 1) Perché hai deciso di seguire Corso FI-ODL? = *Mi interessavano gli argomenti trattati nel corso*
- 1) Perché hai deciso di seguire Corso FI-ODL? = *Mi interessava avere materiale aggiuntivo sulla materia*
- 1) Altri motivi = xxx
- 2) La tua motivazione prima di iniziare il corso era: = *Alta*
- 3) La qualità del materiale didattico fornito nel corso è: = *Molto buona*
- 4) Credi che siano utili per l'apprendimento i test di autovalutazione? = *Molto*
- 5) Ritieni che, in generale, ciascuna lezione sia stata: = *Della giusta lunghezza*
- 5) Eventuali commenti più specifici = xxx
- 6) Esempi e prove proposte sono , in numero e qualità: = *Molto buoni*
- 7) Le prove intermedie sono state: = *Difficili ma autoistruttive*
- 8) Le informazioni ricevute dai tutors durante il corso sono state: = *Più che sufficienti*
- 9) Ritieni che le informazioni e gli aggiornamenti siano pervenuti in tempi: = *Veloci*
- 10) La frequenza con cui hai visitato le pagine del sito è stata: = *Più volte al giorno*
- 10) Altro= xxx
- 11) Durante il corso, la tua richiesta di informazioni è stata rivolta più frequentemente a: = *Docenti*
- 12) Quanto sostegno hai trovato nei tuoi docenti? = *Più che sufficiente*
- 13) Quanto sostegno hai trovato nei tuoi colleghi? = *Scarso*
- 14) Durante il corso, hai avuto la sensazione di sentirti isolato? = *Si, un pò*
- 15) Ritieni che sia stato facile adattarti a questo tipo di corso? = *Abbastanza*
- 16) Il corso che stai valutando è stato tenuto via internet. Considerando i contenuti e le attività svolte, ritieni che un tale tipo di corso sia: complessivamente migliore di uno analogo svolto in modo tradizionale? = *Non so*

17) Quali sono stati i vantaggi nel seguire un corso di questo tipo? = *Scegliere il momento in cui studiare le lezioni ed eseguire le attività connesse*

17) Altro = *vale anche il primo punto, visto che lavoro. Al punto 16) ho risposto "non so" perché ci sono sicuramente vantaggi e svantaggi difficili da valutare vista la novità rappresentata da tale corso*

18) Ritieni che un corso a distanza possa sostituire uno tradizionale? = *In molti casi*

19) Se hai risposto in molti casi o in pochi casi alla domanda precedente, puoi indicarci quali corsi potrebbero essere, organizzati via internet e quali no? = *immagino quasi tutti con le opportune variazioni, es: matematica, elettrotecnica ... fisica no (o solo in parte)*

20) Pensi che questo corso abbia arricchito le tue competenze? = *Si*

21) Rispetto alle tue aspettative, il corso è stato: = *Migliore*

22) Il tuo giudizio complessivo su questo corso è: = *Molto buono*

23) In futuro, seguiresti nuovamente un corso di didattica aperta a distanza? = *Si*

24) Dovendo essere tu a dare un voto complessivo al corso ed ai suoi docenti, quale sarebbe da 1 a 30 ? = *30*

25) Osservazioni e commenti: = *ho trovato molto utili ed efficaci i test di autovalutazione, in quanto vale il detto: sbagliando si impara ... e per chi ha poco tempo a disposizione per studiare, sono un valido aiuto.*

9 Appendice C - Lezione XVI e commenti sulla sua organizzazione

Riportiamo in questa appendice la lezione XVI come si presenta nel sito web <http://digilander.libero.it/capano/index.html>, inserendo alcune immagini esemplificative della stessa e degli strumenti di interazione a disposizione dei learners. Inoltre vi è l'indicazione dell'etichetta del Learning Object

$LOB_{c, j_c, m_c, l_c, o_c, m_c, l_c, m_c, j_c}$ corrispondente al Learning Object di tipo domanda/risposte del test di autovalutazione 16.⁷⁷

9.1 Ordinamento e ricerca

Ordinare e ricercare dei dati secondo determinati criteri è una delle operazioni più importanti nell'ambito dell'elaborazione elettronica.

Tratteremo in questa lezione dell'ordinamento degli array secondo gli algoritmi di selezione e bubble sort e della ricerca degli elementi nell'array secondo il metodo sequenziale ed attraverso la ricerca binaria.

⁷⁷ Vedere l'appendice D

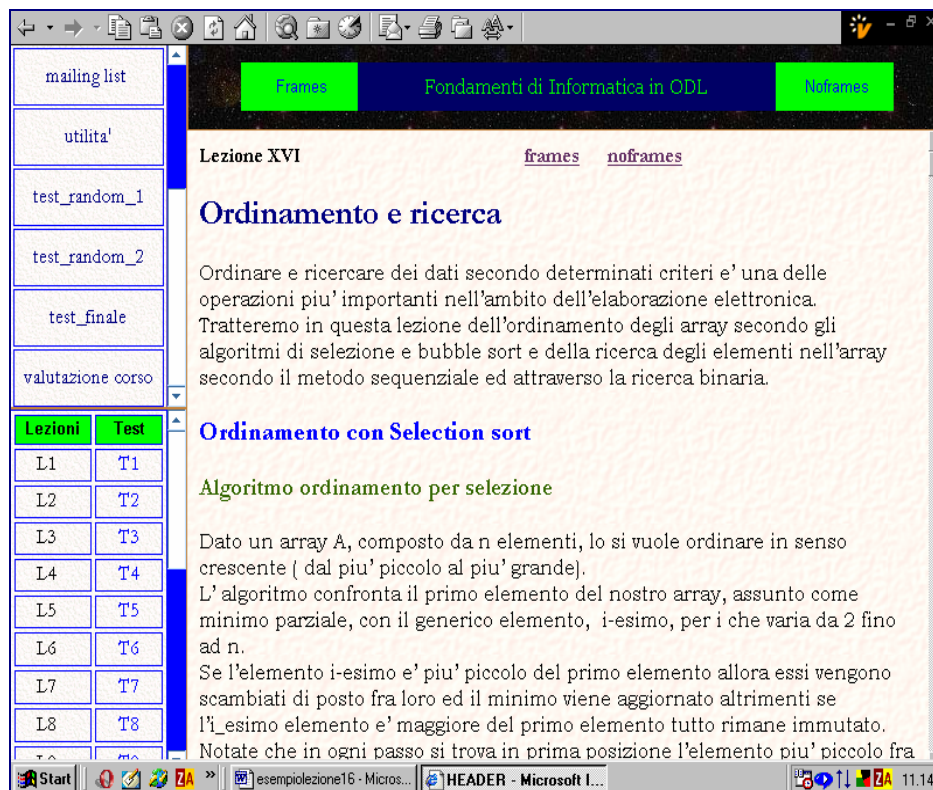


Fig. 9.1 - (Videata Lezione XVI)

9.2 Ordinamento Con Selection Sort

9.2.1 Algoritmo ordinamento per selezione

LO1,2,4,1,2 □ Dato un array A, composto da n elementi, lo si vuole ordinare in senso crescente (dal più piccolo al più grande).

L'algoritmo confronta il primo elemento del nostro array, assunto come minimo parziale, con il generico elemento, i-esimo, per i che varia da 2 fino ad n.

Se l'elemento i-esimo è più piccolo del primo elemento allora essi sono scambiati di posto fra loro ed il minimo viene aggiornato altrimenti se l'i-esimo elemento è maggiore del primo elemento tutto rimane immutato. Notate che in ogni passo si trova in prima posizione l'elemento più piccolo fra quelli

confrontati, fino al termine di tale iterazione dove l'elemento più piccolo di tutto l'array viene a trovarsi in prima posizione.

A questo punto si esegue la operazione di ordinamento fra gli elementi compresi fra il secondo e l'ennesimo; cioè si esegue un confronto fra il secondo (nuovo minimo) ed il terzo, fra il secondo ed il quarto fino a confrontare il secondo con l'ennesimo; ogni volta che dal confronto si trova un elemento più piccolo si esegue uno scambio di posizione fra gli elementi oggetti del confronto ed un aggiornamento del minimo.

Al termine di questa iterazione avremo che in posizione 2 compare l'elemento più piccolo fra gli n-1 elementi che erano rimasti da confrontare.

Il tutto viene ripetuto per il terzo il quarto ... fino allo (n-1)-esimo elemento. Al termine si avrà che lo i-esimo elemento dell'array ha un valore minore od uguale dell'elemento (i+1)-esimo per $i = 1, 2, 3, \dots, n-1$, ossia alla fine del nostro algoritmo l'array risulta completamente ordinato.

La selezione, quindi seleziona l'elemento con valore minore e lo scambia con il primo elemento. Tra gli n-1 elementi rimasti è poi ricercato quello con valore minore e scambiato con il secondo e così di seguito fino all'ultimo elemento.

LO1,2,4,1,1 □ Consideriamo un **esempio grafico selection sort**

Supponiamo di avere il seguente array di interi $\text{int } A[4] = \{55, 12, 44, 9\}$ e di volerlo ordinare in senso crescente, al crescere dell'indice dell'array.

A			
55	12	44	9
0	1	2	3

Nella iterazione i-esima si assume come valore minimo l'elemento i-esimo dell'array per $i = [0 \text{ fino ad } n-2]$. Ad ogni passo il confronto avviene col minimo elemento, suscettibile di eventuali aggiornamenti nei passi precedenti.

Iterazione

Passo 1) minimo=55; confrontiamo 12 con 55: $\text{if}(12 < 55)$ sì ed allora li scambiamo di posto ed aggiorniamo il valore minimo, avremo quindi

$$A'$$

12	55	44	9
----	----	----	---

Passo 2) nuovo minimo=12; confrontiamo 44 con 12 : $\text{if}(44 < 12)$ no ed allora non avviene nessuno scambio

$$A''$$

12	55	44	9
----	----	----	---

Passo 3) minimo=12; confrontiamo 9 con 12 : $\text{if}(9 < 12)$ si ed allora scambiamo i due elementi fra loro:

$$A'''$$

9	55	44	12
---	----	----	----

Nella prima iterazione dopo n-1 passi abbiamo ordinato il primo elemento dell'array essendo nel nostro caso $n=4$ ci sono occorsi 3 passi.

2) Iterazione

Passo 1) minimo=55; confrontiamo il 44 con il 55 : $\text{if}(44 < 55)$ si ed allora scambiamo i due elementi fra loro:

$$A''''$$

9	44	55	12
---	----	----	----

Passo2) minimo=44; confrontiamo il 12 con il 44 : $\text{if}(12 < 44)$ si ed allora scambiamo i due elementi fra loro:

$$A'''''$$

9	12	55	44
---	----	----	----

Nella seconda iterazione dopo n-2 passi abbiamo ordinato il secondo elemento dell'array ed essendo $n=4$ avremo che sono serviti soltanto 2 passi per tale ordinamento

3) Iterazione

Passo 1) minimo=55; confrontiamo il 44 con il 55 : if(44<55) si ed allora scambiamoli fra loro

A''''''

9	12	44	55
---	----	----	----

A questo punto l'array è ordinato, avendovi impiegato 6 passaggi.
Sono serviti 3 passi nella prima iterazione 2 passi nella seconda iterazione ed 1 passo nella terza iterazione, perciò in totale $3+2+1=6$ passi.
Ad ogni passo è eseguito sempre un confronto mentre non è detto sia eseguito lo scambio.

In generale occorrono $(n-1)$ iterazioni;
Nella prima iterazione occorrono $n-1$ confronti; nella seconda iterazione occorrono $n-2$ confronti ; nella terza iterazione $n-3$ confronti eccetera fino alla $(n-1)$ -esima iterazione dove serve un solo confronto.

Per cui il numero di confronti totali che è pari al numero totale di passi dell'algoritmo è dato dalla seguente formula:

$(n-1) + (n-2) + (n-3) + \dots + 2 + 1$ che è uguale a $n * (n-1)/2$

ossia coincide con la formula di Gauss che calcola la somma dei primi $n-1$ numeri interi positivi.

Nel nostro esempio $n=4$, ergo possiamo sapere a priori che il numero di confronti o passi totali che l'algoritmo richiede è pari a:

$$n*(n-1)/2 = 4*(4-1)/2 = 6.$$

Si suole dire in letteratura che l'algoritmo ha un costo dato da $n*(n-1)/2$; ossia la complessità dell'algoritmo dipende dal numero n di elementi da ordinare ed ha sempre lo stesso costo per ogni possibile configurazione dei dati di ingresso. Notiamo pure che anche se l'array fosse ordinato sono effettuati lo stesso $n*(n-1)/2$ confronti.

Essendo $n*(n-1)/2 = (n^2 - n)/2$ si ha che il costo dell'algoritmo ordinamento per selezione è quadratico, ossia cresce all'incirca con il quadrato di n .

LO1,2,4,1,4 □

Una possibile pseudo_codifica dell'algoritmo selection sort è la seguente:

selection_sort (A):

per ogni $i=1,2,\dots,n$ ripeti:

trova l'elemento minimo nel vettore A e ponilo in posizione i -esima

end.

Una possibile rappresentazione in linguaggio C di una funzione che usa tale algoritmo è la seguente,
in cui A è l'array di interi passato come parametro ed n è il numero di interi letti, anche esso passato come parametro :

```
void selection_sort(int A[ ], int n)78
{
    int i, j, i_minimo;
    for ( i=0; i<n-1; i++) /*iterazione*/
    {
        /*ricerca della componente minima fra A[i] e A[n]*/
        i_minimo=i;
        for(j=i+1; j<n; j++) /*passi*/
            if(A[j]<A[i_minimo]) /*confronto*/
                i_minimo= j;
        scambia(A[j],A[i_minimo] );79
    }
}
```

Esercizio

Cliccando sul link **Esercizio** nel Sito compare la finestra con l'esercizio assegnato come compito, riguardante la parte di lezione che si era studiata

⁷⁸ int A[]; passato come parametro alla funzione selection-sort lo avremmo potuto sostituire con int *A "puntatore all'array";

e' evidente che dobbiamo fornire anche come parametro la dimensione dell'array, poiche' nel passaggio di un array ad una funzione questa riceve l'indirizzo di memoria relativo al primo elemento dell'array.

L'indirizzo non fornisce nessuna informazione riguardo al numero di elementi presenti nel vettore e quindi e' compito del programmatore fornire tale numero alla funzione.

⁷⁹ scambia(A[j],A[i_minimo]) naturalmente e' una funzione C da progettare, da parte del programmatore, facendo uso di una variabile di appoggio.

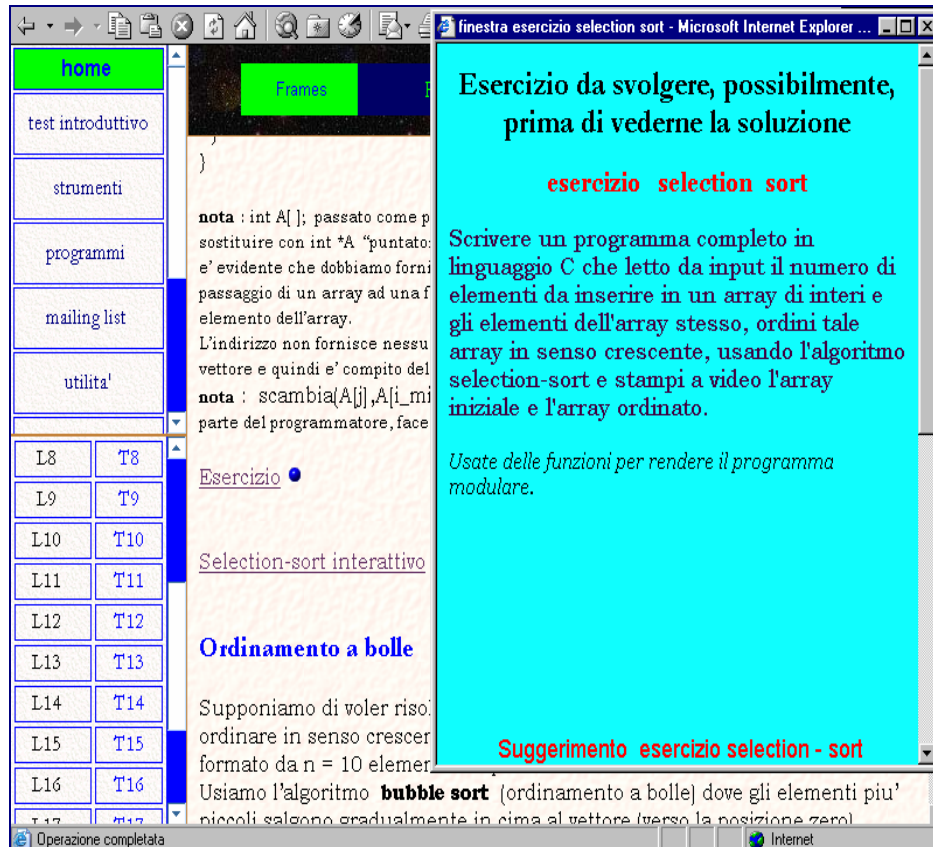


Fig. 9.2 C - (Traccia esercizio selection sort)

Suggerimento

All'interno della finestra, dell'esercizio proposto, cliccando su **Suggerimento esercizio selection-sort** si apre la pagina con il suggerimento dato per svolgere la soluzione. Ci si aspetta da ciò che gli studenti si autodisciplinino, ossia che tentino di risolvere l'esercizio da soli e diano uno sguardo al suggerimento dato ad una possibile sua soluzione, dopo aver provato da soli a risolverlo.

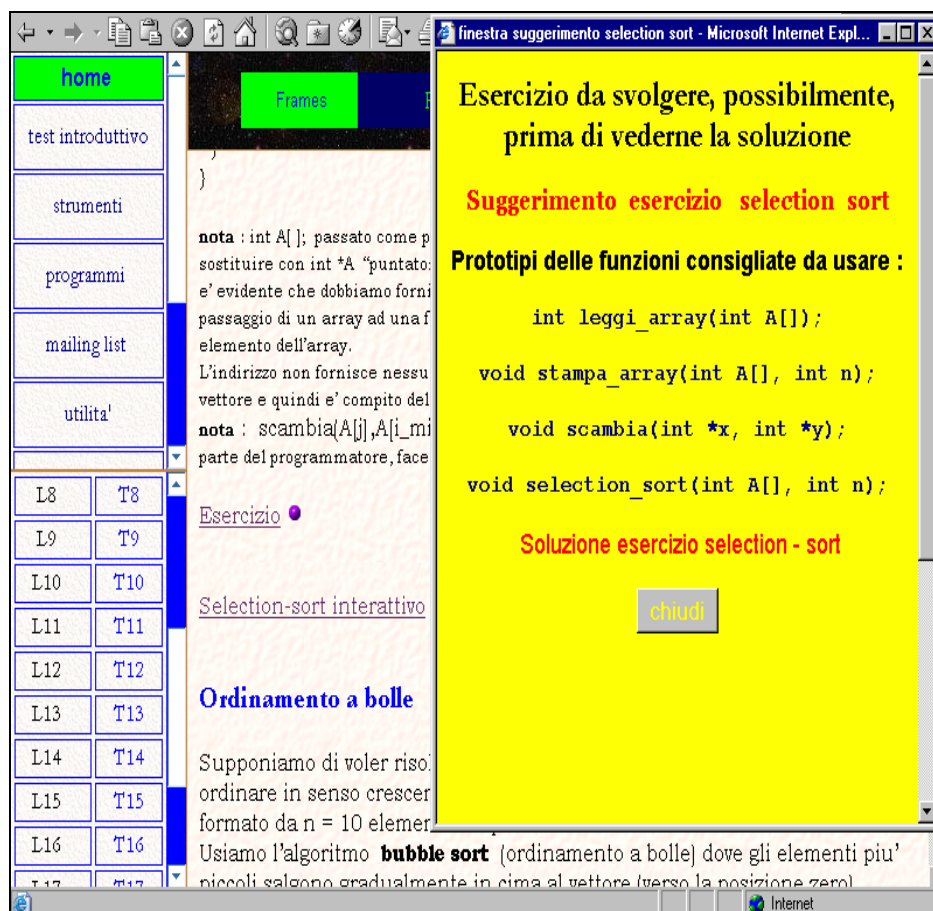


Fig. 9.3 C - (suggerimento esercizio selection sort)

Soluzione

Cliccando sul link del sito **Soluzione esercizio selection-sort** si apre la finestra con la soluzione da noi proposta. Si auspicava che fosse guardata dagli studenti soltanto dopo aver provato da soli a risolvere l'esercizio.

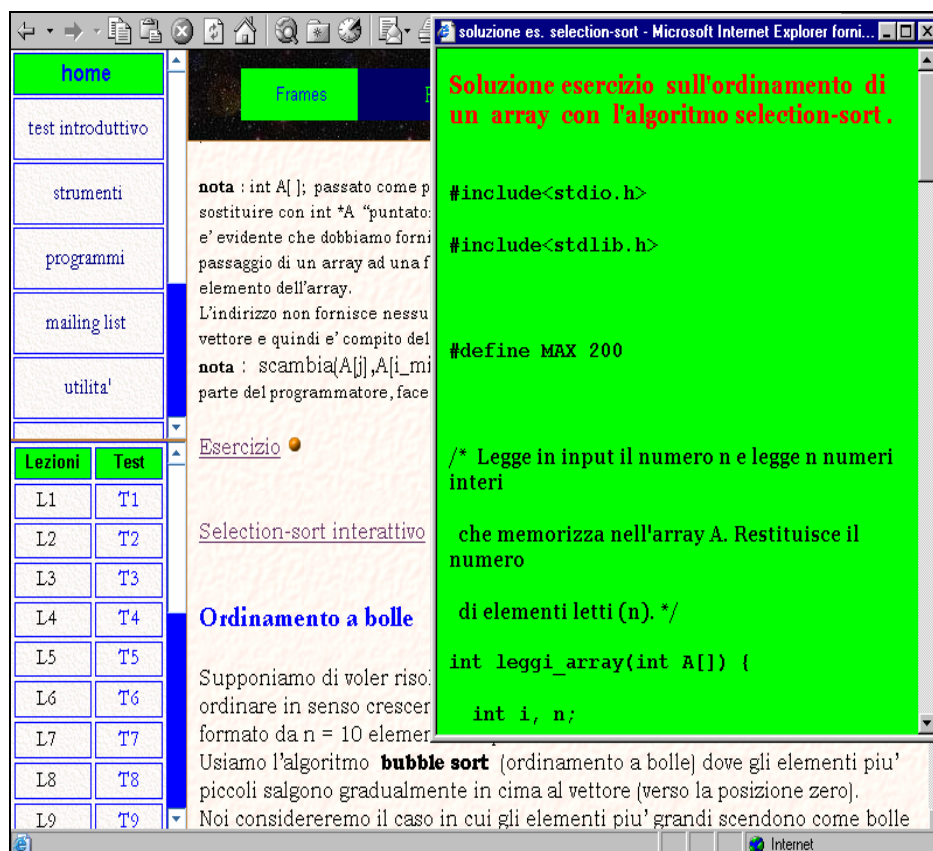


Fig. 9.4 C - (soluzione proposta esercizio selection sort)

80

⁸⁰ In tale finestra è riportata la soluzione dell'esercizio proposta dal tutor. Tale soluzione attraverso il learning cooperativo e collaborativo che si sviluppa nell'aula virtuale della piattaforma, da noi utilizzata, è sicuramente perfezionabile. Peraltro un corso di programmazione non avrà quasi mai soluzioni uniche per cui a maggior ragione in questo caso il learning cooperativo aiuta a migliorare la qualità della didattica (Nuove soluzioni create dagli studenti).

LO1,2,4,1,9 extra □ Selection-sort interattivo

Cliccando sul link del sito **Selection-sort interattivo** si apre la seguente finestra:

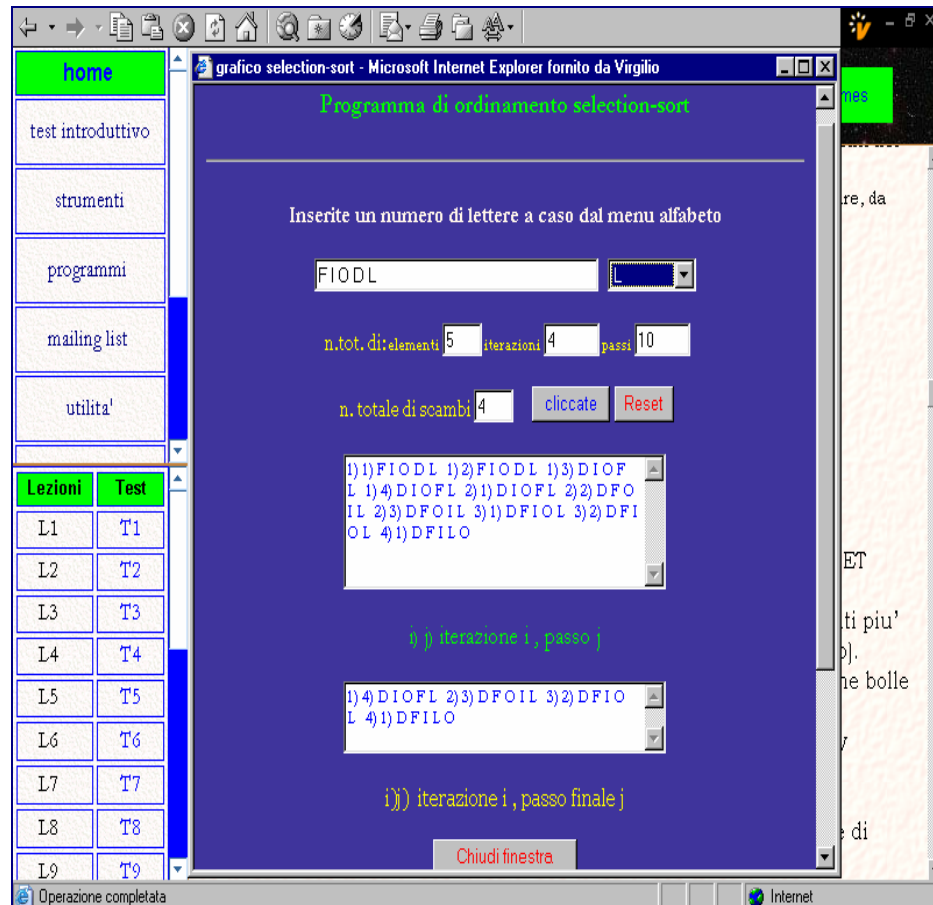


Fig. 9.5 C - (esempio di ordinamento interattivo con selection sort)

In questa finestra si può, interattivamente, verificare il comportamento dell'algoritmo selection sort nell'ordinare una sequenza di caratteri. Dal menu a tendina si selezionano i caratteri che si vuole ordinare, nella textarea superiore appaiono, dopo aver cliccato su **cliccate**, tutte le iterazioni ed i passi che compie l'algoritmo selection-sort nell'ordinare la parola immessa nel campo text.

Nella textarea inferiore si può visualizzare il passo finale ad ogni iterazione. Per implementare l'algoritmo abbiamo utilizzato delle funzioni in Javascript. Riportiamo di seguito il codice completo HTML e le funzioni Javascript della pagina in questione:

```
<html>
<head>
<title>grafico selection-sort</title>
<script language="javascript">
var alfa=new Array();
var cont_lettere=0; //variabile globale
var cont_j=0;
var cont_i=0;
var aux=new Array(2);

function iniz()
{
  alfa.length=0;
  cont_lettere=0; // *inizializzazione del contatore delle lettere*
  cont_i=0;
  cont_j=0;
}

function passa()
{
  if (document.bubble.alfabeto.selectedIndex>0)
  {
    document.bubble.tf1.value=document.bubble.tf1.value +

document.bubble.alfabeto.options[document.bubble.alfabeto.selectedIndex].text
+ " ";
  }

  alfa[cont_lettere]=document.bubble.alfabeto.options[document.bubble.alfabeto.selectedIndex].text ;
  cont_lettere++;
  document.bubble.totale_elementi.value=cont_lettere;
  document.bubble.totale_iterazioni.value=cont_lettere-1;
  document.bubble.totale_passi.value=(cont_lettere)*(cont_lettere-1)/2;
```

```

    }

function passo_selection_sort()
{
var cnt_j=0;
var cnt_i=0;
var passo_scambi=0;
var min;

for(cnt_i=0;cnt_i <=alfa.length-2;cnt_i++)
{
min=cnt_i;
for(cnt_j=(cnt_i+1);cnt_j<alfa.length;cnt_j++)
{
if(alfa[cnt_j]<alfa[min])
{
aux[0]=alfa[cnt_j];
alfa[cnt_j]=alfa[min];
alfa[min]=aux[0];
passo_scambi++;
}

document.bubble.S1.value+=(cnt_i+1)+" ";
document.bubble.S1.value+=(cnt_j+1-cnt_i-1)+" ";
for(i=0;i<alfa.length;i++)
document.bubble.S1.value+=alfa[i]+" ";
document.bubble.S1.value+=" ";
}
document.bubble.S2.value+=(cnt_i+1)+" ";
document.bubble.S2.value+=(cnt_j-cnt_i-1)+" ";
for(i=0;i<alfa.length;i++)
document.bubble.S2.value+=alfa[i]+" ";
document.bubble.S2.value+=" ";
} // end for cnt_i
document.bubble.T2.value=passo_scambi;
}
</script>
</head>

<body>
<p align="center"><b><font size="4">Programma di ordinamento

```


[illegible]

LO1,2,4,1,3 □

9.3 Ordinamento a bolle

Supponiamo di voler risolvere un semplice problema di ordinamento: ordinare in senso crescente, i valori contenuti in un vettore di nome VET formato da $n = 10$ elementi di tipo intero.

Usiamo l'algoritmo bubble sort (ordinamento a bolle) dove gli elementi più piccoli salgono gradualmente in cima al vettore (verso la posizione zero).

Noi considereremo il caso in cui gli elementi più grandi scendono come bolle verso la posizione finale del vettore (posizione $n-1$ esima)

Esso si basa sul principio che presi due elementi adiacenti in un array ordinato, $VET[i]$ e $VET[i+1]$, si ha che

$VET[i] \leq VET[i+1]$.

Con l'algoritmo bubble sort vengono confrontate ripetutamente coppie di elementi adiacenti, le quali vengono scambiate se non rispettano l'ordinamento. Precisamente l'elemento $VET[0]$ viene confrontato con l'elemento $VET[1]$ e se risulta che $VET[0] > VET[1]$ i due elementi vengono scambiati; poi si confronta $VET[1]$ con $VET[2]$ ed in generale l'elemento $VET[i]$ con l'elemento $VET[i+1]$ finché i vale: $n - 1 - \text{numero_di_iterazione_attuale}$.

Durante questa prima iterazione gli elementi maggiori risalgono come bolle verso le posizioni di indice più alto del vettore; avremo che l'elemento di valore massimo si porta in posizione $n - 1 = 9$ nel nostro caso.

Nella seconda iterazione avremo che l'elemento massimo fra quelli rimasti raggiungerà la posizione ($n-2$ esima) e così fino alla ultima iterazione dove avremo che ogni elemento avrà raggiunto la posizione definitiva e quindi il vettore è ordinato. Notiamo che alla prima iterazione effettuiamo ($n - 1$) confronti alla seconda iterazione effettuiamo ($n - 2$) confronti alla terza effettuiamo ($n - 3$) confronti e così via fino alla ($n-1$ esima) iterazione dove si effettua un solo confronto.

In totale avremo $[(n - 1) + (n - 2) + (n - 3) + \dots + 1] = [n * (n - 1) / 2]$ confronti.

Nel caso del nostro vettore VET con $n=10$, saranno necessari 45 confronti e di scambi affinché lo si ordini. Tutto questo avviene considerando il caso peggiore, ossia un vettore con nessun elemento in posizione corretta, in altri termini un vettore ordinato in senso decrescente al crescere dell'indice dell'array. Se l'array ha un ordinamento casuale il numero di scambi è imprecisato.

Vediamo allora il programma C che ordina in modo crescente, al crescere dell'indice dell'array, i valori di un vettore:

```
#include<stdio.h>
#define DIM 10

int main( )
{
    int VET[DIM] = {2, 5, 6, 34, 11, 44, 33, 89, 68, 36 };
    int i, iterazione, vecchio;
    printf ("\nStampa del vettore iniziale da ordinare");
    for (i=0; i<DIM; i++)
        printf ("%d", VET[i] );
    for (iterazione=1; iterazione<=DIM -1; iterazione++) /*Iterazioni*/
        for (i=0; i<=DIM - iterazione -1 ; i++) /*Passi*/
            if (VET[i] > VET[i+1] ) /*confronto*/
                /* scambio*/
                vecchio = VET[i];
                VET[i] = VET[i+1];
                VET[i+1] = vecchio;
        }
    printf ("\nStampa del vettore ordinato in senso ascendente\n");
    for (i=0; i<=DIM-1; i++)
        printf ("%4d", VET[i]);
    printf ("\n");
    return 0;
}
```

LO1,2,4,1,5 □

Consideriamo adesso un **esempio grafico** che mostra il **comportamento dell'algoritmo bubble-sort**.

Supponiamo di avere il seguente array di interi int VET[4]={55,12,44,9} e di volerlo ordinare in senso crescente al crescere dell'indice del vettore.

VET

55	12	44	9
0	1	2	3

Iterazione 1

Passo 1) confrontiamo 12 con 55 ; if(55>12) si ed allora li scambiamo di posto avremo quindi:

VET'

12	55	44	9
----	----	----	---

Passo 2) confrontiamo 55 con 44; if (55>44) si ed allora li scambiamo di posto

VET''

12	44	55	9
----	----	----	---

Passo 3) confrontiamo 55 con 9; if(55>9) si ed allora li scambiamo di posto.

VET'''

12	44	9	55
----	----	---	----

Abbiamo concluso così la prima iterazione ed abbiamo posto l'elemento massimo del vettore nell'ultima posizione dell'array (n-1 esima).

Su quest'indice e sull'elemento contenuto in corrispondenza di tale indice non si lavora più nelle iterazioni successive.

VET''''

12	44	9	55
----	----	---	----

Iterazione 2

Passo 1) confrontiamo il 12 con il 44; if(12>44) no ed allora tutto rimane come prima.

VET'''''

12	44	9	55
----	----	---	----

Passo 2) confrontiamo il 44 con il 9; if (44>9) si ed allora li scambiamo di posto

VET''''''

12	9	44	55
----	---	----	----

Abbiamo concluso i passi della seconda iterazione e portato l'elemento massimo fra gli n-1 elementi su cui stavamo lavorando nella sua posizione corretta la n-2 esima.

Adesso sui due elementi ordinati non si lavora più nelle successive iterazioni.

VET'''''

12	9	44	55
----	---	----	----

Iterazione 3

Passo 1) si confronta il 12 con il 9; if (12>9) si ed allora li scambiamo di posto

VET''''''

9	12	44	55
---	----	----	----

Con quest'iterazione si conclude l'ordinamento dell'array ed abbiamo l'array seguente ordinato in senso crescente al crescere degli indici.

VET

9	12	44	55
---	----	----	----

Esercizio

Nel sito, cliccando su Esercizio, avvengono cose simili a quelle che abbiamo su illustrato.

LO1,2,4,1,10 extra □

9.4 Ordinamento di altre strutture dati

Fino ad adesso abbiamo considerato due algoritmi che operavano l'ordinamento sugli array di interi, lo stesso si potrebbe fare se gli array fossero di tipo (char, float ...); basterebbe soltanto cambiare il tipo di dati degli argomenti e delle variabili delle funzioni usate nell'ordinamento. Al fine di adattare gli algoritmi di ordinamento ad altre strutture dati, basta modificare il codice, in coerenza con la nuova struttura dati coinvolta nell'ordinamento, fermo restando che gli algoritmi rimangono quelli introdotti.

Bubble-sort interattivo

Nel sito cliccando sul link Bubble-sort interattivo compare la finestra della seguente figura:

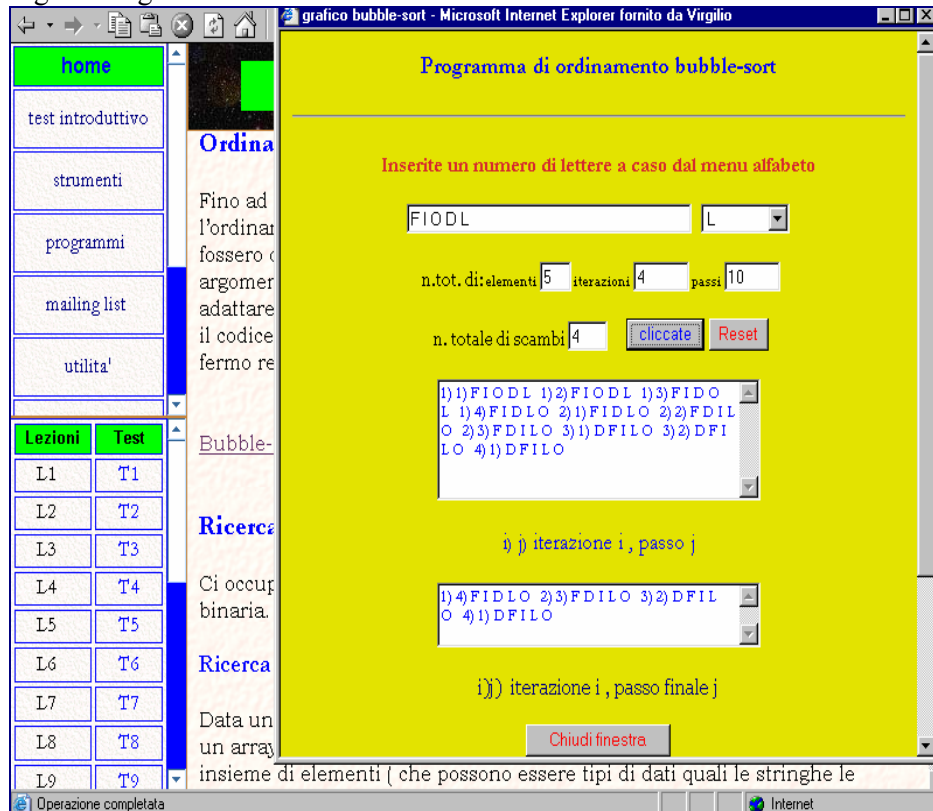


Fig. 9.6 C - (Finestra interattiva Bubble Sort)⁸¹

Anche in questo programma interattivo on-line è possibile vedere il comportamento dell'algoritmo B.S. inserendo lettere da noi scelte nel menu a tendina alfabeto.

⁸¹ Nota abbiamo inserito la medesima parola, FIODL, del nostro esempio in Fig. 9.5 C per meglio poter apprezzare le differenze fra i due algoritmi di ordinamento sopra illustrati.

9.5 Ricerca di un elemento in un array di interi

Ci occuperemo soltanto di due tipi di ricerca la ricerca sequenziale e la ricerca binaria.

LO1,2,4,1,7 □

9.5.1 Ricerca sequenziale

Data una certa chiave (elemento da cercare), si vuole trovare tale elemento in un array di interi od in generale le informazioni relative a tale elemento in un insieme di elementi (che possono essere tipi di dati quali le stringhe le strutture le liste eccetera ..).

In genere questo tipo di ricerca si adotta quando l'array non è ordinato e consiste nello scandire tutto l'array e confrontare ogni suo elemento con la nostra chiave.

Si intuisce che la complessità o costo, nel caso peggiore, di un algoritmo che implementi la ricerca sequenziale è pari al numero di elementi dell'array ossia l'algoritmo ha un costo lineare con n.

Scriviamo adesso una funzione che ricerca un certo carattere in un array di caratteri di lunghezza nota, finché non è soddisfatto il confronto con una chiave specificata.

```
int ricerca_sequenziale(char *elemento, int cont, char chiave)
{
    register int k; /*se può memorizza k in un registro*/
    for(k=0; k<cont; k++)
        if(chiave==elemento[k] )
            return(k); /* ritorna l'indice dove si trova la chiave cercata*/
    else return( -1); /*l'elemento non e' stato trovato*/
}
```

notiamo che la se la chiave cercata è in ultima posizione o non esiste proprio l'array di caratteri elemento viene scandito fino in fondo, e questo dimostra che il costo della ricerca sequenziale nel caso peggiore è pari al numero n di elementi dell'array.

Nel caso medio la ricerca sequenziale effettua un numero di confronti (if (...)) pari ad n/2 e nel caso migliore 1 solo confronto: "elemento cercato si trovava in prima posizione nell'array".

Se si hanno grandi quantità di dati, questo algoritmo diventa inefficiente, per il tempo che si impiegherebbe nel caso in cui l'elemento da cercare capiti in posizione finale nel posto in cui sono immagazzinati tali dati. Si deduce da ciò che bisogna adottare in tali casi altri tipi di algoritmi per effettuare la ricerca in tempi più brevi.

Esercizio

9.5.2 Ricerca binaria in un array di numeri interi

LO1,2,4,1,8 □

L'algoritmo di ricerca binaria si applica soltanto agli insiemi ordinati, secondo un determinato criterio.

Supponiamo di avere un array di nome RB costituito dai primi 15 numeri interi positivi.

RB

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

Se volessimo ricercare l'elemento 11 (chiave) nell'array RB ci comportiamo nel modo seguente:

Il numero da cercare si può trovare in uno dei seguenti posti:

- elemento mediano dell'array ;
- nella metà a sinistra dell'elemento mediano;
- nella metà a destra dell'elemento mediano.

Quindi attraverso la ricerca binaria si confronta 11 con l'elemento mediano 8 if ($11 \geq 8$) si allora sicuramente la nostra chiave 11 si trova nella metà a destra dell'elemento mediano del nostro array o coincide con l'elemento mediano stesso. Se coincide con l'elemento mediano abbiamo finito la ricerca, altrimenti, scartiamo in tale modo dalla ricerca la metà degli elementi, quelli a sinistra di 8.

Adesso troviamo il nuovo elemento mediano della sottometà dell'array a destra dell' elemento mediano che avevamo considerato prima.

RB'

9	10	11	12	13	14	15
---	----	----	----	----	----	----

Il numero da cercare si può trovare nuovamente in uno dei seguenti posti:
elemento mediano dell'array ;
nella metà a sinistra dell'elemento mediano;
nella metà a destra dell'elemento mediano.

Confrontiamo il numero da cercare 11 con il nuovo elemento mediano, il 12
 $\text{if}(11 \geq 12)$ no allora sicuramente l'elemento cercato coincide o con
l'elemento mediano oppure con un elemento situato nella metà a sinistra di
RB'. Visto che non coincide con l'elemento mediano cerchiamo nella
sottometà.

RB''

9	10	11
---	----	----

Il numero da cercare si può trovare nuovamente in uno dei seguenti posti:
elemento mediano dell'array ;
nella metà a sinistra dell'elemento mediano;
nella metà a destra dell'elemento mediano.

Confrontiamo il numero da cercare 11 con il nuovo elemento mediano, il 10
 $\text{if}(11 \geq 10)$ si allora sicuramente l'elemento cercato coincide o con
l'elemento mediano oppure con un elemento situato nella metà a destra di
RB''.

RB'''

11

Il numero da cercare si può trovare nuovamente in uno dei seguenti posti:
elemento mediano dell'array ;
nella metà a sinistra dell'elemento mediano;
nella metà a destra dell'elemento mediano.

Confrontiamo il numero da cercare 11 con il nuovo elemento mediano, il 11
 $\text{if}(11 \geq 11)$ si allora sicuramente l'elemento cercato coincide o con
l'elemento mediano oppure con un elemento situato nella metà a destra di
RB'''. In questo caso coincide con l'elemento mediano e quindi la ricerca è
finita avendo trovato l'elemento che volevamo cercare.

Vediamo adesso di formalizzare meglio il metodo empirico descritto, applicandolo ad un generico array a di n numeri interi.

a

a_0	a_1	a_2	a_3	..	..	a_{med}	..	..	..	a_{n-2}	a_{n-1}
-------	-------	-------	-------	----	----	-----------	----	----	----	-----------	-----------

Detto a_{med} l'elemento mediano dell'array a avremo che il modo per calcolarlo potrebbe essere : $a_{med}=n/2$; ma in tale modo non riusciamo a calcolare l'indice dell'elemento centrale ogni volta che riduciamo gli elementi da confrontare; il metodo che funziona nel caso generale e' il seguente:

$$a_{med} = (estr_sup + estr_inf)/2$$

$estr_sup$ è l'estremo superiore dell'array mentre $estr_inf$ è l'estremo inferiore dell'array.

Infatti ogni volta che si effettua un confronto abbiamo visto che viene eliminata una metà degli elementi dal proseguo della ricerca; per cui gli estremi inferiori e superiori cambiano di volta in volta e necessitano di essere aggiornati contestualmente.

Se la chiave x che stiamo cercando è presente nell'array a significa che deve essere compreso fra $a[estr_inf]$ ed $a[estr_sup]$, di conseguenza necessitano di un aggiornamento in modo che venga rispettata la condizione di presenza dell'elemento x nell'array.

Introduciamo allora il seguente test: $if(x \geq a[a_{med}])$.

Se questa condizione e' vera, x starà nell'intervallo

$[a_{med}, estr_sup]$, quindi si aggiorna : $estr_inf=a_{med}$; altrimenti $estr_sup=a_{med}$;

questo procedimento deve essere sempre iterato fino a quando non risulti che $(estr_inf + 1) = estr_sup$; in tal caso vuol dire che abbiamo controllato tutti gli elementi.

* Descrizione discorsiva dell'algoritmo di ricerca binaria

Definire l'array ed il valore cercato x

Assegnare ai limiti dell'array le due variabili $estr_inf$ ed $estr_sup$

Calcolare la posizione mediana della rimanente porzione di array da esaminare

Ripetere

Se il valore cercato x è maggiore del valore corrente allora
 Aggiorna di conseguenza il limite inferiore
 Aggiorna di conseguenza il limite superiore
 Calcola di nuovo e per ogni passo l'elemento amed
 Se l'elemento a[amed] è uguale all'elemento cercato x allora
 Comunica di averlo trovato (restituisce ad es. l'indice dell'array)
 Altrimenti
 Comunica di non averlo trovato (restituisce ad es. -1)

Scriviamo adesso la versione in linguaggio C della funzione di ricerca binaria:

```

int ricerca_binaria( int a[ ], int x, int n)
{
  int estr_inf=0;
  int estr_sup=n-1;
  int amed ;
  while (estr_inf <= estr_sup)
  {
    amed= (estr_inf + estr_sup)/2;
    if( x>a[amed])
      estr_inf = amed+1;
    else if(x<a[amed])
      estr_sup = amed -1;
    else return (amed);
  }
  return -1;
}

```

LO1,2,4,1,6 □

La complessità dell'algoritmo di ricerca binaria dipende dal particolare elemento da ricercare.

Nel caso si cerchi l'elemento mediano, basta soltanto un solo confronto mentre se si cerca un elemento diverso da quello mediano bisogna effettuare un numero maggiore di confronti (il confronto rappresenta l'operazione dominante e quindi assimiliamo la complessità col numero di confronti).

Il caso peggiore si verifica quando l'elemento che si cerca non è presente nell'array. Dopo k confronti la dimensione dell'array su cui effettuiamo la ricerca si riduce a $n/(2^k)$; pertanto nel caso peggiore il numero di confronti coincide con il più piccolo intero m che soddisfa la seguente relazione:

$2^m \geq n$; da cui: $\log_2 (2^m) \geq \log_2 n$ ed implica $m \geq \log_2 n$

quindi non appena il numero di confronti m è maggiore di $\log_2 n$ la ricerca termina; l'emme che verifica ciò è: $m = P.i.i.(\log_2 n) + 1$; (dove $P.i.i.()$ abbiamo indicato la parte intera inferiore dell'argomento fra parentesi tonde). Possiamo concludere dicendo che la complessità dell'algoritmo di ricerca binaria è funzione del logaritmo in base due di n .

Esempio

Avendo un array di 1025 elementi, ordinati in senso crescente, nel caso peggiore per trovare un elemento dell'array o per scoprire che non esiste, con la ricerca binaria, occorrono $m = P.i.i.(\log_2 1025) + 1$ confronti ossia $m = 11$ confronti.

Esercizio ricerca binaria

The screenshot shows a web browser window with the following elements:

- Navigation Menu (Left Sidebar):**
 - programmi
 - mailing list
 - utilita'
 - test_random_1
 - test_random_2
 - test_finale
 - valutazione corso
 - L13 T13
 - L14 T14
 - L15 T15
 - L16 T16** (highlighted)
 - L17 T17
 - L18 T18
 - L19 T19
 - L20 T20
 - L21 T21
- Page Header:**
 - Frames
 - Fondamenti di Informatica in ODL
 - NoFrames
 - [Lo sai quale è la stella più vicina alla terra?](#)
- Page Title:** finestra esercizio sulla ricerca binaria - Microsoft Internet Explorer fornito da FastWeb
- Main Content Area:**
 - esercizio ricerca binaria**
 - Esempio**
Avendo un array di 1025 elementi, ordinati in senso crescente, nel caso peggiore per trovare un elemento dell'array o per scoprire che non esiste, con la ricerca binaria, occorrono $m = P.i.i.(\log_2 1025) + 1$ confronti ossia $m = 11$ confronti.
 - Esercizio**
Inserite nel programma sulla selection sort, la funzione di ricerca binaria, che cerchi nell'array un elemento passato come parametro, e stampi a video l'array iniziale, l'array ordinato e l'indice dell'elemento eventualmente trovato.
 - Utilizzate il programma selection sort inserendovi soltanto la funzione di ricerca binaria ed alcune righe opportune nel main.
 - Soluzione esercizio sulla ricerca binaria**
 - chiudi** (button)

Fig. 9.7 - (Traccia Esercizio ricerca binaria)

The screenshot shows a web browser window with the following components:

- Sidebar (Left):** A table of contents with links like 'programmi', 'mailing list', 'utilita'', and a list of lessons (L13-T13 to L21-T21). The 'valutazione corso' section is expanded.
- Main Content Area:**
 - Header: 'Fondamenti di Informatica in ...'
 - Text: '... ricerca termina, l'indice che verifichiamo è $\lfloor \log_2 n \rfloor + 1$; (dove $\lfloor \cdot \rfloor$ abbiamo indicato la parte dell'argomento fra parentesi tonde).
 - Text: 'Possiamo concludere dicendo che la complessità di ricerca binaria è funzione del logaritmo in base 2 di n.'.
 - Esempio:** 'Avendo un array di 1025 elementi, ordinati in modo crescente, nel caso peggiore per trovare un elemento che non esiste, con la ricerca binaria, moltiplichiamo il numero di confronti per 10 (ossia $m = \lceil \log_2 1025 \rceil + 1$ confronti)'. Below this is a link 'Esercizio'.
 - Test 16:** A button with a rainbow arrow pointing up.
 - Footer: 'Stampa Lezione 16'.
- Right-Hand Pane:** Titled 'soluzione es. ricerca binaria - Microsoft Internet Explorer', it contains a C program:


```
#include<stdio.h>
#include<stdlib.h>

#define MAX 200

/*
Legge in input il numero n
e legge n numeri interi
che memorizza nell'array A.
Restituisce il numero
di elementi letti (n).
*/

int leggi_array(int A[]) {
    int i, n;

    printf("\nInserite il
numero di elementi
dell'array: ");
    scanf("%d", &n);
    for (i=0; i<n; i++) {
```

Fig. 9.8 - (Soluzione esercizio ricerca binaria)

Test 16

9.6 Videata del link del sito Test 16

Il test di autovalutazione 16 è composto da 8 domande a risposta multipla (Learning Objects di tipo domanda/risposte). Le domande sono sugli argomenti (Learning Objects del SottoModulo di tipo Lezione) studiati nella lezione 16 corrispondente.

The screenshot shows a web browser window with the address bar displaying http://digilander.libero.it/capano/test_singoli/TEST16.htm. The browser's title bar indicates the page is titled 'Fondamenti di Informatica in ODL'. The page content is divided into a sidebar and a main area.

Sidebar (Left):

- home
- test introduttivo
- strumenti
- programmi
- mailing list
- utilita'
- L9 T9
- L10 T10
- L11 T11
- L12 T12
- L13 T13
- L14 T14
- L15 T15
- L16 T16 (highlighted)
- L17 T17

Main Content Area:

Test di autovalutazione 16

Domanda 1 :

Per ordinare il vettore, di Fig 1, con l'algoritmo selection sort, in ordine crescente al crescere dei suoi indici, quanti confronti occorrono :

Fig.1

44	32	29	11	9	7
0	1	2	3	4	5

☐ 12 confronti

☐ 4 confronti

☐ 10 confronti

☒ 15 confronti

Fig. 9.9 C - (Test di autovalutazione 16)

9.6.1 Test di autovaluzione 16, dopo aver risposto a tutte le domande esattamente.

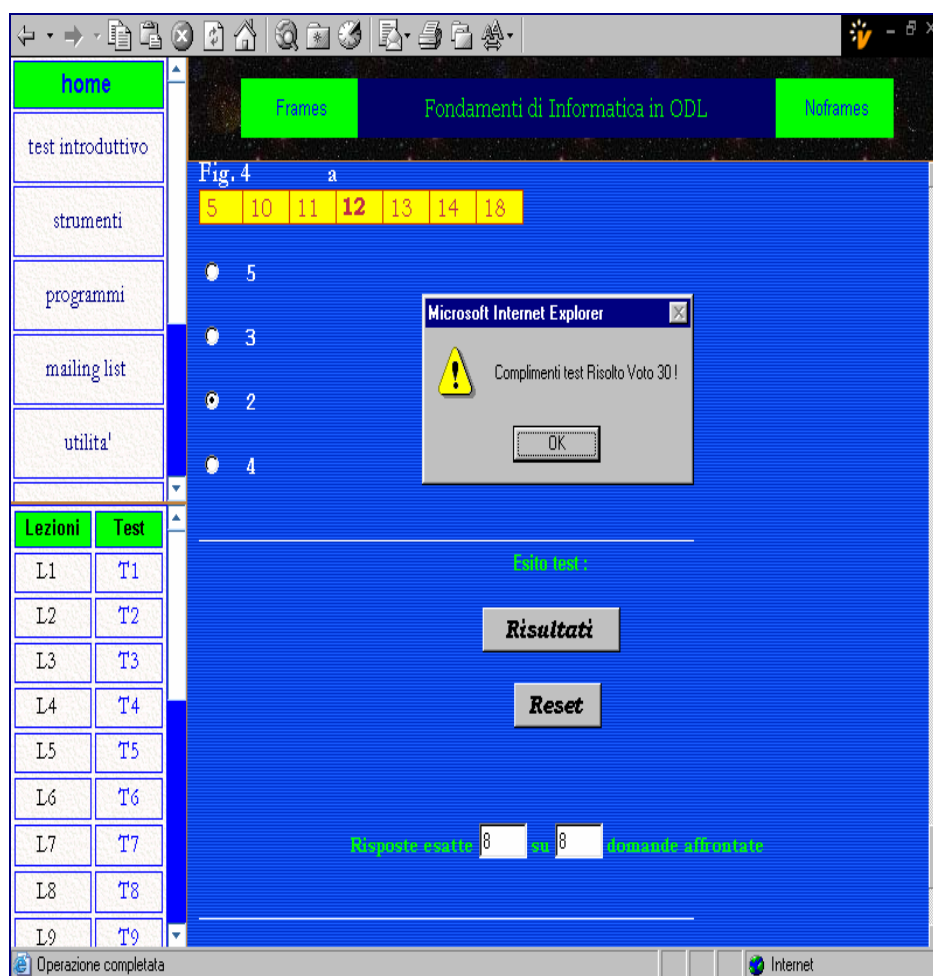


Fig. 9.10 C - (Test di autovaluzione 16 risolto)

9.6.2 Test di autovalutazione 16 con tempo impiegato per la sua risoluzione

Cliccando su OK, dell'alert in mostrata nell figura 9.10 C, si apre un'altra alert che ci indica il tempo impiegato per rispondere alle domande del test 16.

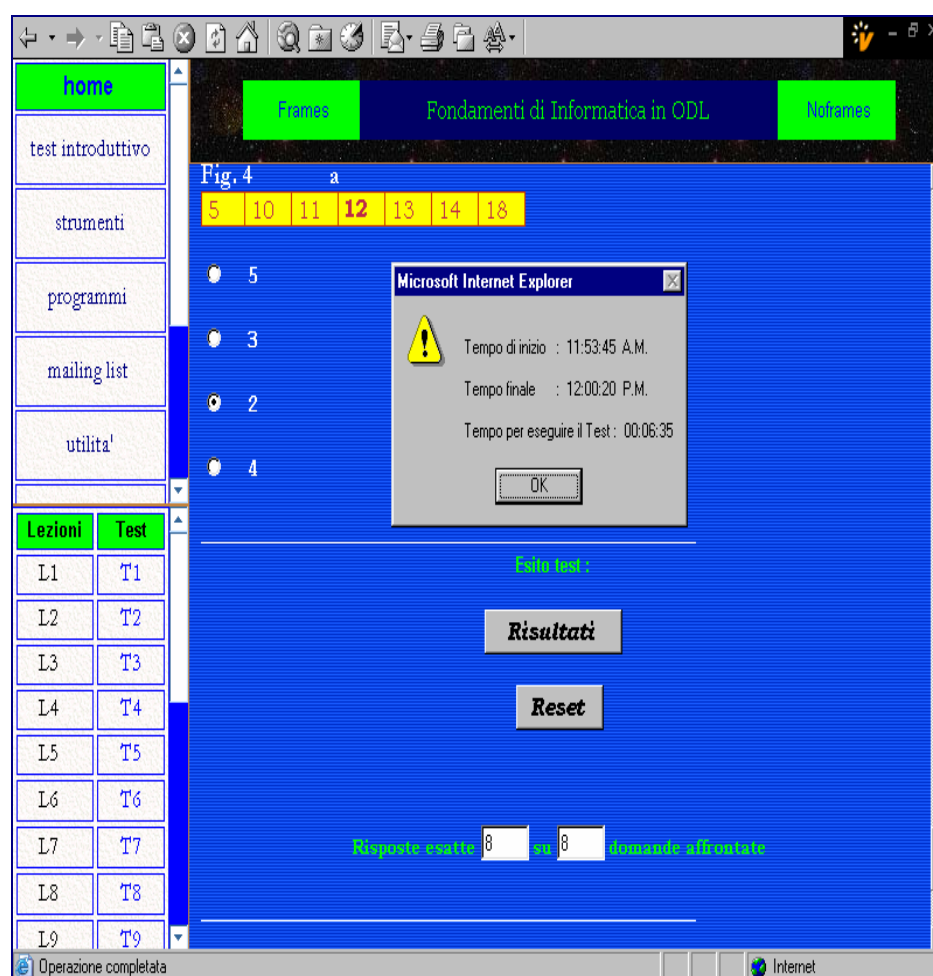


Fig. 9.11 C - (Test di autovalutazione 16 tempo impiegato)

I voti riportati erano normalizzati in trentesimi con possibilità di non necessariamente rispondere a tutte le domande.

10 Appendice D

10.1 I Learning Objects della Lezione 16

Nel capitolo 1 abbiamo diviso il Modulo 1,2,4 in tre SottoModuli rappresentanti rispettivamente la Lezione L16, il Test di Autovalutazione T16 e la Prova Intermedia 3 per F2.

Riprendendo la Tabella 1.5 e le definizioni che abbiamo introdotto di Learning Object di tipo domanda/risposte dei test di autovalutazione e la teoria con cui si è associato ad ogni L.O. di tipo domanda/risposte un Learning Object nel corrispondente SottoModulo di tipo Lezione, esplicitiamo in Tabella 10.1 i L.O. e le corrispondenze didattiche.

Tab. 10.1 - I learning Objects del Modulo 4 di F2

SMod1,2,4,1 ≡ (Lezione 16)	Contenuto del Learning Object del Smod1,2,4,1	SMod1,2,4,2 ≡ (Test autov. 16)	SMod1,2,4,3 ≡ (Prova Intermedia 3 per F2)
L01,2,4,1,1	Esempio Grafico per Selection sort	L01,2,4,2,1	L01,2,4,3,1= prova3/e possibili soluzioni
L01,2,4,1,2	Algoritmo di ordinamento per selezione	L01,2,4,2,2	
L01,2,4,1,3	Ordinamento a bolle	L01,2,4,2,3	
L01,2,4,1,4	Codifica C dell'algoritmo selection sort Ed esercizio	L01,2,4,2,4	
L01,2,4,1,5	Esempio Grafico B.S. ed esercizio	L01,2,4,2,5	
L01,2,4,1,6	Complessità algoritmo Ricerca Binaria	L01,2,4,2,6	
L01,2,4,1,7	Ricerca sequenziale ed esercizio	L01,2,4,2,7	
L01,2,4,1,8	Algoritmo Ricerca Binaria ed esercizio	L01,2,4,2,8	

Il Learning Object, del SottoModulo di tipo Test di Autovalutazione, “domanda/risposte” LO1,2,4,1,1, coincidente con la domanda 1 e le possibili opzioni risposta, del Test di autovalutazione 16 ha un corrispondente Learning Object LO1,2,4,1,1 nel SottoModulo di tipo Lezione. Idem per gli altri 7 Learning Object. Vediamo adesso le corrispondenze.

Il learning Object, del SottoModulo 1,2,4,2, LO1,2,4,2,1 è il seguente:

LO1,2,4,2,1 Domanda 1 : Per ordinare il vettore, di Fig 1, con l'algoritmo selection sort, in ordine crescente al crescere dei suoi indici, quanti confronti occorrono : Fig.1 44 32 29 11 9 7 0 1 2 3 4 5 ☐ 12 confronti ☐ 4 confronti ☐ 10 confronti ☐ 15 confronti

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,1 contenuto nel SottoModulo 1,2,4,1, LO1,2,4,1,1 ha la seguente scheda:⁸²

Etichetta: LO1,2,4,1,1 Requisiti: conoscenza algoritmo selection sort. Misura del tempo di apprendimento: 20 minuti Conoscenza che fa acquisire : Esempio Grafico per Selection sort Descrizione contenuto: <pre><script> esempio(str) {searchWin= window.open(str,'esempio','scrollbars=yes,resizable=yes,width=550, height=500,status=no,location=no,toolbar=no'); }</pre>
--

⁸² Per vedere il Learning Object fisicamente presente nel SottoModulo Lezione 16 andare all'appendice C.

</script>
< codice HTML del contenuto didattico di LO1,2,4,1,1 >
Risorse:BTN-TBAL.GIF

Il learning Object, del SottoModulo1,2,4,2, LO1,2,4,2,2 è il seguente:

LO1,2,4,2,2

Domanda 2 :
Volendo ordinare il vettore in Fig. 2, in senso crescente al crescere degli indici del vettore, con l'algoritmo selection-sort , al secondo passo della seconda iterazione, quale vettore semi-ordinato avremo.

Fig. 2

44	32	29	11	9	7	4
0	1	2	3	4	5	6

a)
O 4 9 29 11 44 32

b)
O 4 7 29 44 9 7 11

c)
O 4 29 44 32 11 9 7

d)
O 32 44 11 29 7 4 9

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,2 contenuto nel SottoModulo1,2,4,1, LO1,2,4,1,2 ha la seguente scheda:

Etichetta: LO1,2,4,1,2
Requisiti: Conoscenza funzioni e gestione Array.
Misura del tempo di apprendimento: 15 minuti
Conoscenza che fa acquisire: Algoritmo di ordinamento per selezione
Descrizione contenuto:
< codice HTML del contenuto didattico di LO1,2,4,1,2>
Risorse: Nessuna

Il learning Object, del SottoModulo1,2,4,2, LO1,2,4,2,3 è il seguente:

LO1,2,4,2,3

Domanda 3 :

Il seguente programma cosa fa all'array VET?

```
#include<stdio.h>
```

```
#define N 10
```

```
int main( )
```

```
{
```

```
int VET[N] = {2, 5, 6, 34, 11, 44, 33, 89, 68, 36 };
```

```
int i, k, vecchio;
```

```
for (i=0; i<DIM; i++)
```

```
printf (“\n%d”, VET[i] );
```

```
for (k=1; k<N ; k++)
```

```
for (i=0; i<=N -1 ; i++)
```

```
if (VET[i] > VET[i+1] )
```

```
{
```

```
vecchio = VET[i];
```

```
VET[i] = VET[i+1];
```

```
VET[i+1] = vecchio;
```

```
}
```

```
return 0;
```

```
}
```

☐ lo ordina in senso decrescente, solo parzialmente, con il bubble-sort

☐ lo ordina in senso crescente utilizzando il bubble-sort

☐ lo ordina in senso decrescente utilizzando la selection-sort

☐ lo ordina in senso crescente utilizzando la selection-sort

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,3 contenuto nel SottoModulo1,2,4,1, LO1,2,4,1,3 ha la seguente scheda:

Etichetta: LO1,2,4,1,3

Requisiti: Conoscenza gestione Array.

Misura del tempo di apprendimento: 20 minuti

Conoscenza che fa acquisire: Ordinamento a Bolle

Descrizione contenuto : < codice HTML contenuto didattico di LO1,2,4,1,3 >

Risorse: nessuna

Il learning Object, del SottoModulo1,2,4,2, LO1,2,4,2,4 è il seguente:

LO1,2,4,2,4

Domanda 4 :

La funzione seguente cosa fa all'array A passatole come parametro essendo n il numero di elementi dell'array A ?

```
void selection_sort(int A[ ], int n)
{
    int i, j, i_minimo;
    for ( i=0; i<n-1; i++) /*iterazione*/
    {
        /*ricerca della componente minima fra A[i] e A[n]*/
        i_minimo=i;
        for(j=i+1; j<n; j++) /*passi*/
            if(A[j]<A[i_minimo]) /*confronto*/
                scambia(A[j],A[i_minimo] );
    }
}
```

☐ non riesce ad ordinarlo in senso crescente poiché manca l'istruzione i_minimo= j; dopo il confronto.

☐ lo ordina in senso crescente correttamente

☐ non riesce ad ordinarlo in senso crescente poiché il confronto dovrebbe essere if(A[j]>=A[i_minimo])

☐ lo ordina in senso decrescente correttamente

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,4 contenuto nel SottoModulo1,2,4,1, LO1,2,4,1,4 ha la seguente scheda:

Etichetta: LO1,2,4,1,4

Requisiti: Conoscenza funzioni e gestione Array.

Misura del tempo di apprendimento: 15 minuti

Conoscenza che fa acquisire: Codifica C dell'algoritmo Selectio Sort

Descrizione del contenuto:

<SCRIPT>
esempio2(str)

```
{searchWin=
window.open(str,'esempio','scrollbars=yes,resizable=yes,width=550,
height=500,status=no,location=no,toolbar=no');
}
</SCRIPT>
< codice HTML del contenuto didattico di LO1,2,4,1,4 >
```

Risorse:BTN-TBAL.GIF

Il learning Object, del SottoModulo1,2,4,2, LO1,2,4,2,5 è il seguente:

LO1,2,4,2,5

Domanda 5 :

Volendo ordinare il vettore in Fig. 3, in senso crescente al crescere degli indici del vettore, con l'algoritmo bubble-sort , al secondo passo della seconda iterazione quale vettore semi-ordinato avremo.

Fig. 3

| | | | | | | |
|---|----|----|----|---|---|----|
| 4 | 32 | 29 | 11 | 9 | 7 | 44 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

- ☐ 4 32 11 29 7 44 9
- ☐ 4 11 29 9 7 32 44
- ☐ 32 44 11 29 7 4 9
- ☐ 4 9 11 29 7 32 44

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,5 contenuto nel SottoModulo1,2,4,1, LO1,2,4,1,5 ha la seguente scheda:

Etichetta: LO1,2,4,1,5

Requisiti: Conoscenza funzioni e gestione Array.

Misura del tempo di apprendimento: 20 minuti

Conoscenza che fa acquisire: Esempio Grafico Bubble Sort ed esercizio su B.S..

Descrizione contenuto:

```
<SCRIPT>
esempio2(str)
{searchWin=
window.open(str,'esempio','scrollbars=yes,resizable=yes,width=550,
height=500,status=no,location=no,toolbar=no');
}
</SCRIPT>
< codice HTML del contenuto didattico di LO1,2,4,1,5>
```

Risorse: nessuna

Il learning Object, del SottoModulo1,2,4,2, LO1,2,4,2,6 è il seguente:

LO1,2,4,2,6

Domanda 6 :

Avendo un array di $(2^{20} + 1)$ interi, ordinato in senso decrescente al crescere degli indici, quanti confronti occorrono nel caso peggiore, con la ricerca binaria (adattata a tale caso) per trovare un elemento, che sappiamo essere in esso presente.

- ☐ almeno 50 confronti
- ☐ al massimo 15 confronti
- ☐ 21 confronti
- ☐ 51 confronti

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,6 contenuto nel SottoModulo1,2,4,1, LO1,2,4,1,6 ha la seguente scheda:

Etichetta: LO1,2,4,1,6**Requisiti:** Conoscenza funzioni e gestione Array.**Misura del tempo di apprendimento:** 10 minuti**Conoscenza che fa acquisire:** Complessità algoritmo Ricerca binaria**Descrizione contenuto:**

< codice HTML del contenuto didattico di : LO1,2,4,1,6>

Risorse: freccia_su.gif

Il learning Object, del SottoModulo1,2,4,2, LO1,2,4,2,7 è il seguente:

LO1,2,4,2,7

Domanda 7 :

A quale compito assolve la seguente funzione funz:

int funz (int elemento[], int cont, int chiave)

```
{
register int k;
for(k=0; k<cont; k++)
    if(chiave==elemento[k] )    return(k);
else return( -1); }
```

O restituisce l'indice dell'array in cui e' contenuto l'intero cercato con la ricerca sequenziale oppure -1 se non trova l'intero richiesto

O restituisce l'elemento dell'array in cui e' contenuto l'intero cercato, con la ricerca sequenziale, oppure -1 se non trova l'intero richiesto

O restituisce l'indice dell'array in cui e' contenuto l'intero cercato con la ricerca binaria oppure -1 se non trova l'intero richiesto

O restituisce l'elemento dell'array in cui e' contenuto l'intero cercato, con la ricerca binaria, oppure -1 se non trova l'intero richiesto

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,7 contenuto nel SottoModulo1,2,4,1, LO1,2,4,1,7 ha la seguente scheda:

Etichetta: LO1,2,4,1,7

Requisiti: Conoscenza funzioni e gestione Array.

Misura del tempo di apprendimento: 20 minuti

Conoscenza che fa acquisire: Ricerca Sequenziale ed esercizio su R.S.

Descrizione contenuto:

<SCRIPT>

esempio2(str)

{searchWin=

window.open(str,'esempio','scrollbars=yes,resizable=yes,width=550,height=500,status=no,location=no,toolbar=no');

}

</SCRIPT>

< codice HTML del contenuto didattico di LO1,2,4,1,7>

Risorse: BTN-TBAL.GIF

Il learning Object, del SottoModulo1,2,4,2, LO1,2,4,2,8 è il seguente:

LO1,2,4,2,8

Domanda 8 :

Volendo cercare l'elemento 14 in un array, a, così ordinato come in Fig. 4, dopo quanti passi, della istruzione while della ricerca binaria, lo riusciamo ad individuare essendo noti i parametri della funzione di ricerca binaria a,x,n :

Fig. 4

a

5 10 11 12 13 14 18

- ☐ 5
- ☐ 3
- ☐ 2
- ☐ 4

Il corrispondente Learning Object, della domanda/risposte LO1,2,4,2,8 contenuto nel SottoModulo1,2,4,1, LO1,2,4,1,8 ha la seguente scheda:

Etichetta: LO1,2,4,1,8

Requisiti: Conoscenza funzioni e gestione Array, algoritmo di ricerca binaria.

Misura del tempo di apprendimento: 20 minuti

Conoscenza che fa acquisire: Algoritmo di Ricerca Binaria ed Esercizio R.B.

Descrizione contenuto:

```
<SCRIPT>
esempio2(str)
{searchWin=
window.open(str,'esempio','scrollbars=yes,resizable=yes,width=550,
height=500,status=no,location=no,toolbar=no');
}
</SCRIPT>
```

< codice HTML del contenuto didattico di LO,12,4,8>

Risorse: BTN-TBAL.GIF

10.2 I Learning Objects extra della L16

Nella lezione L16 avevamo due L.O. extra, (nel senso che non vi è diretta relazione con le domande di T16), LO1,2,4,2,9 ed LO1,2,4,2,10.

Etichetta: LO1,2,4,1,9

Requisiti: Conoscenza algoritmo Selection Sort

Misura del tempo di apprendimento: 10 minuti

Conoscenza che fa acquisire: Bubble Sort Interattivo

Descrizione contenuto :

```
<SCRIPT>
```

```
esempio(str)
```

```
{searchWin=
```

```
window.open(str,'esempio','scrollbars=yes,resizable=yes,width=550,
```

```
height=500,status=no,location=no,toolbar=no');
```

```
}
```

```
</SCRIPT>,
```

```
<table border="1" width="100%" height="95">
```

```
<tr>
```

```
<td width="100%" height="89">
```

```
<font face="Arial" size="3">
```

```
LO1,2,4,1,9
```

```
<p><A
```

```
href="javascript:esempio('../grafico_ordinamento/grafico_selection_sort.htm')
```

```
interattivo</A><IMG src="BTN-TBAL.GIF"
```

```
border=0 width="14" height="14"></A>
```

```
</p>
```

```
</font>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<codice html di grafico_selection_sort.htm>
```

Risorse: BTN-TBAL.GIF,

Etichetta: LO1,2,4,1,10

Requisiti: Conoscenza algoritmo Bubble Sort

Misura del tempo di apprendimento: 10 minuti

Conoscenza che fa acquisire: Bubble Sort Interattivo

Descrizione contenuto:

```
<SCRIPT>
esempio(str)
{searchWin=
window.open(str,'esempio','scrollbars=yes,resizable=yes,width=550,
height=500,status=no,location=no,toolbar=no');
}</SCRIPT>

<table border="1" width="100%">
<tr>
<td width="100%">
<font face="Arial" size="3">
LO1,2,4,1,10
<p><A
href="javascript:esempio('../grafico_ordinamento/grafico_bubble_sort.htm'
)">Bubble-sort
interattivo</A><IMG src="BTN-TBAL.GIF"
border=0 width="14" height="14"></A>
</p>
</font></td>
</table>

<codice html di grafico_bubble_sort.htm>
```

Risorse: BTN-TBAL.GIF,

11 Appendice E

Allo scopo di permettere una quantificazione del lavoro implementativo svolto nell'ambito dell'esperimento via web, nella sezione seguente elenchiamo il contenuto delle lezioni, il numero di pagine formato A4 di ogni lezione, gli esercizi di valutazione proposti ed i test di autovalutazione presenti in ogni singola lezione, nonché la tabella che elenca numero di domande e tipo test autovalutazione. Successivamente, per completezza, diamo un esempio di soluzione proposta agli studenti relativamente alla terza Prova Intermedia per Fondamenti 2 ed anche il codice HTML e JavaScript usato per inviare E-mails a tutta la mailing list.

11.1 Contenuto delle lezioni ed attività didattiche correlate

LEZIONE 1: Architettura dei Calcolatori elettronici (PARTE I)

Introduzione

Schema funzionale di un calcolatore

Le Unità di Ingresso e di Uscita

La Memoria Secondaria

Parentesi sul concetto di Algoritmo

1 Esercizio finale, con soluzione proposta

1 Test di Autovalutazione finale di 5 domande a risposta multipla

Numero di pagine della Lezione: 5

LEZIONE 2: Architettura dei Calcolatori elettronici (PARTE II)

La Macchina di Von Neumann

La Memoria Centrale del Calcolatore

Schema dell'Unità di Memoria Centrale

Numero di pagine della lezione: 4

LEZIONE 3: Architettura dei Calcolatori elettronici (PARTE III)

Funzionamento dell'Unità di elaborazione: Come lavora la CPU

Il linguaggio Macchina (PARTE 1)

1 Esercizio finale con soluzione

1 Test di autovalutazione finale (5 domande a risposta multipla)

Numero di pagine della lezione: 4

LEZIONE 4: Architettura dei calcolatori, il Linguaggio Macchina (Parte 2)

Sintassi e Semantica di ogni Istruzione

Esempio di Scrittura di un Programma

1 Esercizio finale con soluzione

1 Test di autovalutazione finale (5 domande a risposta multipla)

Numero di pagine della lezione: 7

LEZIONE 5: Architettura dei calcolatori, il Linguaggio Macchina (Parte 3)

Riassunto: Programmi per risolvere problemi

Esempio di Linguaggio macchina simbolico

Tabella delle istruzioni del Linguaggio Macchina Simbolico (LMS)

Complemento alle Lezioni sul Linguaggio Macchina

Esercizio 5.1 svolto passo passo

Esercizio 5.2 svolto passo passo

Esercizio 5.3 svolto passo passo. 1 Esercizio da svolgere

1 Test di autovalutazione finale

Numero di pagine della lezione: 9

LEZIONE 6: Linguaggi di Programmazione, sintassi e semantica

Linguaggi Formali e Naturali

Linguaggi e Grammatiche

Metalinguaggio EBNF extended Bakus-Naur Form

Esempi di Sintassi EBNF applicati al Linguaggio C

Diagrammi Sintattici

1 Esercizio finale svolto

Test di Autovalutazione finale

Numero di pagine lezione: 8

LEZIONE 7: Concetti di Programmazione in Linguaggio C (Parte I)

Struttura generale di un programma

Problema: analisi, sintesi, algoritmo, programma in linguaggio simbolico,
programma in linguaggio macchina

Diagrammi di flusso

Programma in linguaggio pseudo-pascal

Traduzione programma in linguaggio C

Traduzione del programma in linguaggio macchina

Esempio di un programma espresso con i diagrammi di flusso

Traduzione programma in linguaggio C

Il Concetto di Variabile

Cenni su Istruzioni della Parte Esecutiva

1 esercizio da svolgere

Test di autovalutazione finale

Numero di pagine Lezione: 11

LEZIONE 8: Concetti di Programmazione (Parte II)

Risoluzione di un problema con metodo ingenuo
Risoluzione del medesimo problema in linguaggio C
Tecnica del Contatore
Esempi di Test
Struttura di iterazione for
Traduzione in C degli Esempi di Test
Diagramma di flusso e traduzione in C di un problema
Esempio: Calcolo del MCD(m,n) diagramma di flusso
Esempio: Programma corrispondente in Linguaggio C
Ciclo di vita di un programma
Errori Logici
Programmabilità di un algoritmo
Esempi di algoritmi programmabili tradotti in linguaggio C
Test di autovalutazione
Numero di pagine lezione: 12

LEZIONE 9: Rappresentazione dell'informazione (Parte I)

Sistemi di numerazione ed algoritmi di conversione
Sistemi posizionali
Sistema binari
Operazioni aritmetiche
Rappresentazione di un numero con parte frazionaria
Conversione di base
Caso di numeri con parte frazionaria (reali)
Primo caso particolare. Secondo caso particolare
Test di autovalutazione finale
Numero pagine lezione: 9

LEZIONE 10: Rappresentazione dell'informazione (Parte II)

Numeri interi in memoria centrale
Rappresentazione in modulo e segno
Rappresentazione in complemento alla base
Rappresentazione dei caratteri
I numeri reali
Rappresentazione in virgola fissa
Rappresentazione in virgola mobile (floating point)
Forma normalizzata
1 Esercizio svolto
Test di autovalutazione finale
Numero di pagine lezione: 8

LEZIONE 11: Rappresentazione dell'informazione (Parte III)

Rappresentazione in complemento alla base

Operazioni aritmetiche in complemento a 2

L'aritmetica in virgola mobile

Precisione della rappresentazione di un numero reale

Test di autovalutazione finale

Numero di pagine lezione: 10

LEZIONE 12: Sistema di Elaborazione

Memoria secondaria o Memoria di massa

Bootstrap

Sistema operativo

Struttura Hardware-Software del sistema di calcolo

Alcuni comandi del DOS

Fasi della compilazione di un programma

* Analizzatore lessicale * Analizzatore sintattico * Analizzatore di
semantica statica

Grafo per ottenere un programma funzionante

Scrittura di un programma C, Compilazione, Esecuzione

Stesure parziali di un programma C

Stesure finali di un programma C

Esercizio finale con suggerimento e soluzione

Test di autovalutazione finale

Numero pagine Lezione: 15

LEZIONE 13: Le Funzioni

Struttura di una funzione

Due esempi di programmi C

Funzioni void

Regole di visibilità delle funzioni

Argomenti delle funzioni e modalità di passaggio degli argomenti

Esempi di passaggio degli argomenti per valore

Sintassi della definizione di una funzione

La ricorsione

1 Esercizio intermedio con suggerimento e soluzione

1 Esercizio finale con suggerimento e soluzione

Test di autovalutazione finale

Numero di pagine Lezione: 8

LEZIONE 14: Cenni sui puntatori e passaggio per indirizzo nelle funzioni

Puntatori

Operatori sui puntatori ed esempi

Puntatori come argomenti di una funzione (passaggio per indirizzo)

Funzioni che restituiscono un puntatore

Esercizio finale svolto

Test di autovalutazione finale

Numero di pagine lezione: 7

LEZIONE 15: Array e stringhe

Array ad una dimensione

1 esercizio intermedio svolto, 1 esercizio intermedio svolto

Definizione del tipo array

Esercizio intermedio svolto

Inizializzazione degli array mono-dimensionali

Array a due dimensioni

Inizializzazione degli array bi-dimensionali

Risoluzione di un problema

Esercizio intermedio svolto

Calcolo del prodotto di due matrici quadrate

Puntatori e stringhe

Aritmetica dei puntatori

Scansione di un array con i puntatori

Test di autovalutazione intermedio

Passaggio di array a funzioni

Esercizio intermedio svolto

Stringhe

La funzione scanf()

Alcune funzioni di libreria strcpy(s1,s2), strlen(s1), strcmp()

Esercizio intermedio svolto su strcpy, su strlen, su strcmp

Esercizio intermedio sulle stringhe

Array di stringhe

Esercizio finale svolto

Test di autovalutazione finale

Numero di pagine lezione 15: 16

LEZIONE 16: Ordinamento e Ricerca

Algoritmo di ordinamento per selezione (Selection-Sort)

Esempio grafico s.s.

Programma in C selection sort

Esercizio intermedio svolto

Selection sort interattivo

Algoritmo di ordinamento a Bolle

Programma in C bubble sort

Esempio grafico b.s.

Bubble sort interattivo

Ricerca di un elemento in un array di interi

Ricerca sequenziale

Esercizio svolto sulla ricerca sequenziale

Ricerca Binaria

Esempio grafico svolto sulla ricerca binaria

Programma in linguaggio C sulla ricerca binaria

Esercizio finale svolto linguaggio C

Test di autovalutazione 16

Numero pagine della lezione 16: 11

LEZIONE 17: INPUT/OUTPUT e gestione dei file

I file e gli stream

Il puntatore al file

Apertura di un file

Chiusura di un file

Esercizio intermedio svolto

MODALITA' DI ACCESSO AI FILE

Creazione di un file sequenziale, esempi

Funzioni di lettura e scrittura

fgetc, fputc

La funzione rewind()

Le funzioni di lettura e scrittura formattata

Cenni sui file ad accesso casuale

Le funzioni di lettura e scrittura per blocchi, fread, fwrite, fseek

Esercizio intermedio svolto

Test di autovalutazione finale

Numero di pagine lezione: 9

LEZIONE 18: Allocazione dinamica e puntatori a funzioni

Breve introduzione alla macchina virtuale del C

Le funzioni di allocazione dinamica del linguaggio C

Allocazione dinamica degli array

2 Esercizi intermedi svolti

Errori frequenti usando i puntatori

Esempi su funzioni aventi array come argomenti

La funzione realloc()

Esercizio intermedio svolto

Allocazione esatta di stringhe ed array di stringhe

2 esercizi intermedi svolti

I puntatori a funzioni

Esempi di array di puntatori a funzioni

2 Esercizi finali svolti

Test di autovalutazione finale

Numero di pagine lezione: 14

LEZIONE 19: Il tipo di dato struttura

Introduzione

Puntatori a strutture

Passaggio di strutture alle funzioni

Array di strutture

Programma C che utilizza array di strutture e file

Funzioni che ritornano strutture

Esempi svolti

Esercizio finale svolto

Test di autovalutazione

Numero pagine lezione: 17

Lezione 20: Strutture dati, Liste Parte I

Cenni sui tipi di dati astratti

La rappresentazione dei tipi astratti

Le liste, Le liste semplici

Rappresentazione di una lista semplice

Rappresentazione sequenziale

Rappresentazione collegata o concatenata

Realizzazione delle operazioni

Rappresentazione collegata delle liste semplici

Uso di tipi di dati in C per la rappresentazione collegata

Rappresentazione collegata tramite puntatori

Test di autovalutazione finale

Numero di pagine lezione: 13

LEZIONE 21: Strutture dati, Liste Parte II e gli Alberi binari

Varianti della rappresentazione collegata delle liste

Liste composite

GLI ALBERI BINARI

Parte della terminologia relativa agli alberi

Visite di un albero binario

Visita in preordine

Visita in postordine

Visita simmetrica

RAPPRESENTAZIONE DI UN ALBERO BINARIO

Rappresentazione sequenziale

Esempi svolti, Esercizio intermedio svolto

Rappresentazione mediante lista doppia

Rappresentazione mediante struct e puntatori

GLI ALBERI BINARI DI RICERCA

Esercizio finale svolto

Test di autovalutazione finale

Numero di pagine lezione: 13

Tab. 11.1 - Numero di domande in FIODL e tipo test autovalutazione

Test	Numero Domande	Tipo Test
T1	5	Risposta Multipla
T2	5	R.M.
T3	5	R.M.
T4	5	R.M.
T5	5	R.M.
T6	5	R.M.
T7	8	R.M.
T8	8	R.M.
T9	5	R.M.
T10	8	R.M.
T11	8	R.M.
T12	5	R.M.
T13	6	Opzioni
T14	6	Completamento
T15	8	R.M.
T16	8	R.M.
T17	6 dom. random su 15.	R.M.
T18	8	R.M.
T19	8	Opzioni, Completamento
T20	8	R.M.
T21	8	R.M.
Test intermedio	5	R.M.
Test unico	1	Opzioni
Test random Lez. F1	10	R.M. sel. 1 test fra 4
Test random Lez. F1	Selez. 1 test fra 14 test	Misto
Test random Lez. F2	Sel 1 test fra 11 test	Misto
Domande ricapit. 1	24 domande random	R.M.
Domande ricapit. 2	20 domande random	R.M.

11.1.1 Soluzione proposta alla Prova Intermedia 3 di F2

/*Progettate un programma in linguaggio C, che esegua la fusione tra due vettori di interi A e B ordinati in senso crescente, producendo un terzo vettore C ordinato.
Il vettore risultante dalla fusione deve contenere tutti gli elementi di A e di B anche se tali elementi si ripetono*/

```
#include <stdio.h>
#define DIM_A 100
#define DIM_B 100
#define DIM_C 200

/*prototipi delle funzioni usate*/
int leggi_A(int A[]);
int leggi_B(int B[]);
void fusione_vettori(int numa,int numb,int A[],int B[]);

int main()
{
    int A[DIM_A];
    int B[DIM_B];
    char car;
    int na,nb;
    na=leggi_A(A);
    nb=leggi_B(B);
    fusione_vettori(na,nb,A,B);
    printf("\nInserite un carattere\n");
    scanf("\n%c",&car);
    return(0);
}

int leggi_A(int A[])
{
    int i, num_a;
    printf("\nInserite il numero di elementi del vettore A ");
    scanf("%d",&num_a);
    for(i=0;i<num_a;i++)
    {
        printf("inserite il valore %d del vettore A ",i+1);
        scanf("%d",&A[i]);
    }
    for(i=0;i<num_a;i++)
        printf("%d\n",A[i]);

    return(num_a);
}

int leggi_B(int B[])
{

```

```

int i, num_b;
printf("\nInserite il numero di elementi del vettore B ");
scanf("%d",&num_b);
for(i=0;i<num_b;i++)
{
printf("inserite il valore %d del vettore B ",i+1);
scanf("%d",&B[i]);
}
for(i=0;i<num_b;i++)
printf("%d\n",B[i]);

return(num_b);
}

void fusione_vettori(int numa,int numb,int A[DIM_A],int B[DIM_B])
{
int C[DIM_C];
int i=0,j=0,num_c=0;
while((i<numa)&&(j<numb))
{
if(A[i]<=B[j])
{
C[num_c]=A[i];
i++;
num_c++;
}
else
{
C[num_c]=B[j];
j++;
num_c++;
}
}
while(i<numa)
{ C[num_c]=A[i];
i++;
num_c++;
}
while(j<numb)
{C[num_c]=B[j];
j++;
num_c++;
}
printf("\necco il vettore fusione ordinato C se gli elementi di A e B ");
printf("sono stati inseriti ordinati\n");
for(i=0;i<num_c;i++)
printf("%d\n",C[i]);
}

```

11.1.2 Codice per effettuare l'invio a tutta la mailing list

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Inserisci un messaggio </title>
<style>
A:link, A:visited { text-decoration: none }
A:hover { text-decoration: none; color: #C7C901}
STRONG { font-style:normal; font-size:10px}
</style>
<style fprolloverstyle>A:hover {color: #FF0000; text-decoration:
underline}
</style>

<script language="JavaScript">

function Vai()
{
Posta="mailto:"
Nomi = new Array
Nomi[0]=new
Array('fondII.elt.*****@dis.uniroma1.it', 'aga***@tiscalinet.it',
'nap***@oneonline.it', 'pic***fabio@hotmail.com', 'wil***@tiscalin
et.it', 'dad***@tiscalinet.it', 'bel***@inwind.it', 'ran***@yahoo.c
om', 'vin***@tin.it', 'con***@tiscalinet.it');
Nomi[1]=new
Array('mimmocapano@comunedasa.it', 'fed***@tiscalinet.it', 'fab***
@iol.it', 'cost***@libero.it', 'and***@tiscalinet.it', 'fde***@tin.
it', 'aga***@libero.it');

// ecco. all'infinito

oo="?subject="
Oggetto="Mailing List FI-ODL"
bb="&body="
Corpo=document.forms[0].Testo.value;
for (i=0; i<Nomi.length; i++) {
location.href=Posta+Nomi[i]+oo+Oggetto+bb+Corpo;}
}

</script>
</head>
```

```

<body bgcolor="#FFFFFF" text="#000000" link="#0033CC"
vlink="#6699FF" alink="#00CC33">
<div align="center">
  <center><table border="1" cellpadding="0" cellspacing="0"
width="100%" height="100%" bordercolordark="#6699FF"
bordercolorlight="#99CCFF">
    <tr>
      <td width="100%" height="4%" valign="middle"
align="center">

<form method="POST" action="" width="100%" height="100%">
  <div align="center">
    <center><table border="0" cellpadding="0" cellspacing="0"
width="100%" height="100%">
      <tr>
        <td width="100%" height="7%" valign="middle"
align="center"><p align="center"><b><font color="#000080">Invio
E-mail a tutta la mailing list</font></b></td>
      </tr>
      <tr>
        <td width="100%" height="7%" align="center"
valign="middle">
          <p align="center"><font color="#0000FF">Scrivete la mail
da inviare</font></td>
        </tr>
        <tr>
          <td width="100%" height="70%" valign="middle"
align="center"><p align="center"><textarea rows="12"
name="Testo" cols="40" style="word-spacing: 0; background-color:
#FFEEDD; border: 1 solid #00FFFF; margin-left: 0; margin-top: 0;
margin-bottom: 0"></textarea></p>
          <p align="center"><input type="button" value="Invia"
name="Invia" onclick="Vai()"
onMouseOver="this.style.backgroundColor='darkgreen'" onMouseOut
= "this.style.backgroundColor='blue'" style="font-size: 10 pt;
font-family: Arial; color: #00FF00; font-weight: bold"><input
type="reset" value="Reimposta" name="B1" style="font-size: 10
pt; font-family: Arial; color: #FFFF00; font-weight: bold"
onMouseOver="this.style.backgroundColor='cyan'" onMouseOut =
"this.style.backgroundColor='blue'"><input type="button"
value="Chiudi " name="B2" onclick="window.close()" style="font-
size: 10 pt; font-family: Arial; color: #00FFFF; font-weight:
bold" onMouseOver="this.style.backgroundColor='yellow'"
onMouseOut = "this.style.backgroundColor='blue'"></p>
          </td>
        </tr>
      </table></center>
    </div>

```



```
<input type="hidden" name="mode" ><input type="hidden"
name="m" value="309">
</form>
<p align="center">&nbsp;</p>
<p>&nbsp;</td>
</tr>
</table></center>
</body>
</html>
```

• Bibliografia

- E-Learning : Una guida operativa*, Esposito Giuseppe e Giuseppe Mantese; ed. Franco Angeli, Milano (2003)
- Formazione digitale: progettare l'e-learning centrato sull'utente*, Bonani G.P. ed. Franco Angeli, Milano (2003)
- Le Prospettive Didattiche del Semantic Web*, Petrucco C., Atti Didamatica 2003, (2003)
- C Corso completo di programmazione*, Deitel Harvey M. e Paul J. Deitel, ed. Apogeo, (2000).
- Fondamenti di informatica Linguaggio di riferimento C*, Beneventano Domenico, Sonia Bergamaschi e Claudio Sartori ed. Società editrice Esculapio.
- Fondamenti di programmazione dei calcolatori elettronici*, Batini Carlo, L. Carlucci Aiello, M. Lenzerini, A. Marchetti Spaccamela e A. Miola ed. Franco Angeli, (2000).
- Basi per la programmazione in linguaggio C*, Temperini Marco ed. Società editrice Esculapio, (1998).
- JavaScript La guida*, David Flanagan ed. Apogeo (2000).
- Teoria e Progetto di Algoritmi Fondamentali*, Ausiello G., A. M. Spaccamela e M. Protasi, ed. Franco Angeli, (1988)
- C Ansi C e C++*, Schildt Herbert ed. McGraw-Hill, Libri Italia srl.
- Sistemi Elettronici Digitali*, Marietti Piero, G. Ciccarella, A. Londei e F. Loriga; Masson editoriale Esa, Milano (1996)
- Microsoft FrontPage 2000*, Guillemain Thomas, ed. Mondadori, (2000)

• Sitografia

- <http://www.ctu.unimi.it/osservatorioanee>: “e-LEARNING: CONVEGNO STATO DELL'ARTE E PROSPETTIVE DI SVILUPPO I RISULTATI DELL'OSSERVATORIO ANEE 2003 con il patrocinio del Ministro per l'Innovazione e le Tecnologie” – Università degli Studi di Milano – Lunedì, 30 giugno 2003.
- <http://integrazione.mymedia.it/e-learning.pdf>: “e-Learning for all”, di Fabrizio Pecori.
- http://www.isvor.it/linee_guida.pdf: “Criteri per la qualità nell'e-Learning”, di Ezio Fregnan e Federica Garbolino, ISVOR FIAT 2002.
- <http://www.isvor.it/e-learning1.html> a) Garbolino F. “Dalla Fad all'e-learning: la formazione a distanza non è più così distante”, in AA.VV. Imparando. Esperienze e valori di 30 anni, Ed. Aspasia, IAL Emilia Romagna (nov. 2002).
- b) D'Amato, D.; Garbolino F. "Piattaforme di e-learning: non solo corsi on-line". Iter, supplemento al n. 7 (nov. 2002).
- c) Garbolino; Maffei, “Gli errori da evitare in un progetto e-Learning”, art. di Sistemi & Impresa N.4 (maggio 2002)

http://www.blucomfort.com/internetime/tutto/pag_articolo.php?articolo_ID=cas_30:
 “Standard e metadati”, Internet-Time, e-Learning magazine articolo del 26 maggio
 2003 di Paola Perino
<http://europa.eu.int/comm/education/elearning/comit.pdf> : “Metodologia e tecnologia
 per l’e-learning” di Enrico Cavalli, Agostino Lorenzi, Università degli Studi di
 Bergamo.
<http://elearning.unibg.it/doc.htm>
<http://www.uninettuno.it> : Consorzio NETTUNO.
<http://www.elearningtouch.it>: Community e-Learning
<http://wiley.ed.usu.edu/index.html>, Davide Wiley
http://www.crimpa.it/intraserv/documents/AIXIA_2003.pdf: IWT,
 “Una Piattaforma Innovativa per la Didattica Intelligente su Web” di Nicola Capuano,
 Matteo Gaeta e Alessandro Micarelli. Università di Salerno e Università di Roma
<http://www.momanet.it/iwt/> : IWT Intelligent Web Teacher altra descrizione.
<http://www.spaghettilearning.com/> : Portale di Spaghettilearning, è un progetto tutto
 italiano per la creazione di una piattaforma Open Source completamente gratuita
 per la formazione in ambiente Universitario, Scolastico, nella Pubblica
 Amministrazione e nel Business. La piattaforma PHP + MySQL consente di poter
 utilizzare su qualsiasi sistema operativo (Win, Linux, Solaris, MacOS 10, Unix) il
 tool di formazione.
<http://www.mininnovazione.it/ita/index.shtml>: Ministero per l’Innovazione e le
 Tecnologie.
<http://www.affariefinanza.it>: Laura Kiss, “Cresce l’Italia nel mondo dell’e-Learning”,
 Affari&Finanza, (5 aprile 2004)
<http://www.asfor.it>: Glossario e-Learning di Asfor.
<http://www.isi.edu/in-notes/rfc2396.txt>: T. Berners-Lee, R. Fielding, L. Masinter.
 Uniform Resource Identifiers (URI): Generic Syntax.
<http://www.w3.org/2001/sw/> : The Semantic Web Initiative Home Page.
<http://elite.polito.it/courses/IIIliv/>: Laura Farinetti, Fulvio Corno, *L’Architettura del
 Semantic Web*, Dip. Automatica e Informatica Politecnico di Torino.