

Unità di Apprendimento **STRUTTURE DI DATI** (03/09)

area delle Informazioni / 2 media

1. Le strutture di dati (nella mente dell'uomo)

l'organizzazione della conoscenza dichiarativa in strutture di dati

- la conoscenza dei **procedimenti** / raggruppare *sequenze di azioni*
- la conoscenza dei **dati** / raggruppare *dati* relativi ad oggetti

la progettazione e la gestione di archivi

2. Gli archivi

2.1. l'archiviazione: l'organizzazione dei dati

2.1.1. l'**oggetto** a cui si riferiscono le **proprietà**

2.1.2. proprietà: **nome** e **valore**

2.1.3. l'**unità informativa** come raccolta di valori di proprietà riferiti ad un unico oggetto

2.1.4. l'**archivio** come unione di più unità informative nelle quali il valore di una proprietà è uguale

2.1.5. la **regolarità** di un archivio / l'**archivio multiplo**

2.2. l'archiviazione: la definizione degli scopi

2.3. la progettazione di archivi

2.3.1. la tutela della privacy

2.3.2. l'archivio **chiuso** / l'archivio **aperto**

2.3.3. la gestione degli archivi

3. L'accesso ai dati

l'accesso ai dati collocati su un supporto

3.1. l'**accesso sequenziale**

3.1.1. la **lista**

3.1.2. la **struttura ad albero**

3.2. l'**accesso diretto**

3.2.1. l'accesso diretto mediante **chiavi d'accesso**

- la **tavola**

- la **tabella a doppia entrata**

3.2.2. l'accesso diretto mediante **indici**

- il **vettore**

- la **matrice**

4. Strutture di dati nella memoria interna del calcolatore

*l'importanza di chiamare con un unico nome un insieme organizzato di dati (struttura di dati). Le variabili semplici e le variabili complesse o **matrici** (gli **elementi**, gli **indici**)*

4.1.1. la dichiarazione iniziale e il dimensionamento (**dim**), il ridimensionamento (**redim**)

4.2. il **vettore**

4.3. la **matrice bidimensionale**

4.4. le **matrici multiple**

5. Ingresso e uscita verso le memorie secondarie

La memorizzazione di archivi di dati su **file sequenziali**

Il **campo** e il segno di fine campo. Il **record** come raggruppamento logico di campi. Il ruolo dei campi iniziali o **campi di testa**

Il segnale di "fine del file" (**eof**)

6. Archiviare con i data base

6.1. preparazione del data base

6.2. l'importazione dei dati

6.3. l'importanza della regolarità

6.4. la consultazione del data base (la maschera)

6.5. l'uso dei connettivi per cercare con la maschera

6.6. la consultazione del data base (la selezione)

LE STRUTTURE DI DATI*

Come abbiamo già visto, in particolare nell'U. di A. Algoritmi, ognuno di noi per vivere e risolvere i problemi grandi e piccoli che ci capitano ogni giorno ha a disposizione due tipi di conoscenze:

la *conoscenza dei procedimenti* che raggruppa *sequenze di azioni*

la *conoscenza dei dati* che raggruppa tutti i *dati* relativi agli *oggetti*(**) che ci circondano

L'efficienza con cui riusciamo a risolvere i nostri problemi quotidiani dipende dalla nostra capacità di far lavorare insieme questi due tipi di conoscenza.

Ad esempio: anche se ricordiamo bene la procedura per inserire un compact disk di musica leggera e per far funzionare il lettore, difficilmente riusciremo ad ascoltarlo se non ricordiamo dove abbiamo messo il disco che desideriamo ascoltare.

La gestione di strutture di dati con il calcolatore

Volendo utilizzare il calcolatore non solo per la gestione di semplici variabili ma anche di strutture complesse di dati, è importante ricordare che, al contrario della nostra mente, esso non è in grado di collegare le informazioni in base al loro significato. Spetta a noi, quando elaboriamo dei programmi, organizzare i dati in modo logico e funzionale al problema che intendiamo risolvere.

Come già visto nell'U. di A. Algoritmi, nel calcolatore vengono utilizzati dei procedimenti per gestire dei dati. Per farlo organizza la sua memoria RAM in due distinte aree: una serve a memorizzare i programmi, l'altra a memorizzare i dati.

- Il calcolatore accede alla *memoria programmi* in modo *sequenziale*; parte leggendo ed eseguendo la prima istruzione e poi tutte le altre, una dopo l'altra. Il programma presente nella memoria del calcolatore mantiene la stessa forma sequenziale quando viene salvato su una *memoria secondaria* come **file** (file BASIC o file eseguibile).
- L'area che serve a *memorizzare i dati* ha invece un accesso diretto. Ogni spazio di memoria è raggiungibile grazie ad un suo *indirizzo*. E' il calcolatore che, leggendo nel programma il nome di una variabile e conoscendone l'indirizzo, ne raggiunge il contenuto senza dover passare per altri spazi di memoria. Ma anche queste informazioni, quando vengono memorizzate su *memoria secondaria*, dovranno assumere la struttura di un file sequenziale formato da una lista di dati memorizzati in **campi**. Il nome del file servirà a definire l'intero archivio; in genere esso ha il suffisso ".txt".

Prima di occuparci di come il calcolatore può gestire grandi quantità di dati, chiariamo alcuni concetti che riguardano la raccolta e l'archiviazione dei dati da parte dell'uomo.

* in informatica viene definita **struttura di dati** un insieme di dati raggruppati secondo un determinato criterio.

** in informatica viene definito **oggetto** la realtà (cosa, animale, persona, ecc..) alla quale si riferiscono le informazioni raccolte

GLI ARCHIVI

➤ ***l'archiviazione: l'organizzazione dei dati***

Chiariamo dunque che:

- viene definito **oggetto** la realtà (cosa, animale, persona, ecc..) alla quale si riferiscono i dati raccolti. Ogni oggetto ha una serie di caratteristiche che vengono chiamate **proprietà**, che hanno un **nome** e un **valore**. Prendendo ad esempio un alunno della classe avremo i seguenti nomi e valori di proprietà:

nome di proprietà	valore di proprietà
cognome	<i>Rossi</i>
nome	<i>Paolo</i>
data di nascita	<i>27/01/94</i>
città di nascita	<i>Milano</i>
indirizzo	<i>Via Teodosio, 33</i>
scuola frequentata	<i>Quintino di Vona</i>
classe frequentata	<i>2 H</i>
colore dei capelli	<i>biondi</i>
altezza	<i>1.64</i>
ecc ..	<i>...</i>



- viene definita **unità informativa** un insieme di valori di proprietà riferiti ad un unico oggetto. Nell'esempio appena visto tutti i dati segnati nella seconda colonna costituiscono l'unità informativa dell'alunno preso ad esempio.
- viene definito **archivio** l'unione di più unità informative nelle quali il valore di una proprietà è uguale
Se ad esempio realizzo un'unità informativa per ogni alunno della classe e le riunisco, avrò un archivio che avrà come comune valore di proprietà:
classe frequentata *2 H*
A questo archivio darò un nome che in genere si riferisce al valore di proprietà che lo unisce. L'archivio appena realizzato potrà essere chiamato: **alunni di 2 H**.
Viene inoltre definito **regolare** un archivio formato da unità informative che hanno gli stessi nomi di proprietà. La regolarità di un archivio è molto importante se la sua gestione dovrà essere affidata ad un calcolatore.
- infine viene definito **archivio multiplo** (o di secondo livello) l'unione di più archivi.
Ad esempio, se unisco gli archivi degli alunni di 2 H con quelli di 1 H e quelli di 3 H avrò realizzato un archivio multiplo

➤ ***l'archiviazione: la definizione degli scopi***

Ma perché vengono organizzati degli archivi e in base a quale criterio andranno scelte le proprietà da osservare quando viene organizzato un archivio?

....

Possiamo dunque osservare come l'aver a disposizione un archivio di dati sia indispensabile per svolgere molte attività. L'insegnante non riuscirebbe a valutare gli alunni se non avesse a disposizione i dati relativi ai propri alunni, in particolare per quanto riguarda il rendimento scolastico. Un medico non riuscirebbe a curare i propri pazienti se non avesse a disposizione tutti i dati relativi alla loro salute. Dunque in entrambi i casi, oltre ai valori di proprietà che servono ad individuare l'oggetto (cognome, nome, indirizzo, data di nascita, ecc...), vi saranno le proprietà utili alla propria attività.

La scelta delle proprietà dipende dunque dagli scopi di chi organizza l'archivio.

Anche un libro di testo può essere preso in considerazione come oggetto con le sue proprietà (titolo, autore, casa editrice, ecc..).

Organizziamo dunque un archivio su 3 dei nostri libri di testo:

nomi delle proprietà	Libri di (nome dell'archivio)			
	1° libro	2° libro	3° libro	ecc...



La guida settimanale dei programmi TV dal 29 aprile al 5 maggio è un esempio di archivio di 3° livello: per ogni programma televisivo viene realizzata un'unità informativa che ne illustra le principali caratteristiche; le unità vengono raggruppate in un archivio per ogni emittente televisiva. Per ogni giorno si forma così un archivio di 2° livello formato dall'insieme degli archivi delle singole emittenti. Un libretto raggruppa i programmi dei vari giorni della settimana, formando così un archivio di 3° livello.

1 OTTOBRE MERCOLEDÌ

LA 7 ☎ 06 355841

6.00 ● Meteo - Traffico - Tg La 7

7.00 OMNIBUS LA7 attualità

9.35 DUE MINUTI UN LIBRO rubr.

9.40 FA' LA COSA GIUSTA (repl.)

10.35 VITE ALLO SPECCHIO (repl.)

11.25 LA LEGGE DI BURKE
«Pietanze micidiali» tel.

12.30 ● Tg La7/SPORT 7

13.10 LAW & ORDER
«Un figlio modello» tel.

14.15 ● **FILM WESTERN** ◆◆
LAMPI NEL SOLE
1958 con Susan Hayward

16.00 FA' LA COSA GIUSTA

16.55 VITE ALLO SPECCHIO

17.45 HOMICIDE
«Colpo in canna» tel.

18.45 DISCOVERY PRESENTA doc.

19.45 ● Tg La7

20.20 SPORT 7

20.30 OTTO E MEZZO

21.30 SFERA attualità

23.30 ● Tg La 7

24.00 THE STRIP
«Opening night» tel.

1.05 OTTO E MEZZO (repl.)

2.05 DUE MINUTI UN LIBRO rubr. (r.)

2.10 ● Cnn

CINEMA

12.40

14.35

16.10

17.50

19.25

21.00

22.45

0.10

1.50

MTV

CINEMA

Sempre da "Guida TV" abbiamo l'elenco dei programmi de *LA 7* (nome archivio 1° livello) di *mercoledì 1° ottobre* (nome archivio 2° livello) nella *settimana dal 28 sett. al 4 ottobre* (nome archivio di 3° livello).

Osserva l'unità informativa indicata dalla freccia e scrivi qua sotto i valori delle proprietà con i relativi nomi:

14.15 > orario di trasmissione

> _____

> _____

> _____

> _____

> _____

> _____

un valore di proprietà può essere espresso anche senza usare le parole. Ad esempio i segni ◆◆ indicano il giudizio della critica:

◆◆ (discreto) > giudizio della critica
mentre il cerchio verde indica i programmi che possono essere visti da tutti.

SABATO 20 SETTEMBRE 2003

I FILM DEL GIORNO

STARGATE
FANTASTICO, COLORE, 1994, USA, DURATA 119'
Regia di Roland Emmerich. Con Kurt Russell, James Spader, Viveca Lindfors. Le piramidi non sono state costruite dall'uomo, ma da creature aliene. Lo scopre un archeologo americano (James Spader) che, decifrando i simboli di un anello egizio, si apre un passaggio verso un altro mondo. Lo accompagna nell'avventuroso viaggio un colonnello (Kurt Russell) con i suoi uomini.
Italia 1 – 20.30

UN VOLTO DAL PASSATO
THRILLER, COLORE, 1999, USA, DURATA 119'
Regia di Marc S. Grenier. Con Alexandra Paul, Vlasta Vrana. Una donna poliziotto è sulle tracce di un serial killer scatenato. Ha scoperto che sta uccidendo proprio gli uomini accusati di averla violentata qualche anno prima.
RaiDue – 21.00

BRIAN'S SONG
DRAMMATICO, COLORE, 2001, USA, DURATA 89'
Regia di John Gray. Con Sean Maher, Mekhi Phifer, Ben Gazzara. La storia dell'amicizia di due giocatori di calcio, dentro e fuori dal campo, messa a dura prova dalla malattia.
Sky Cinema 3 – 21.00

PRESO DI MIRA
AZIONE, COLORE, 1998, USA, DURATA 95'
Regia di Jerry Jameson. Con Jeff Speakman, William Shatner, Larry Cedar. Un dirigente della campagna elettorale scopre che il candidato per cui sta lavorando è implicato in un complotto politico.
Retequattro – 23.50

IL TERRORE CORRE SUL FILO
THRILLER, B/N, 1948, USA, DURATA 88'
Regia di Anatole Litvak. Con Barbara Stanwyck, Burt Lancaster, Ann Richards. Il marito la vuole uccidere. Lo scopre la ricca e nevrotica Barbara Stanwyck intercettando per caso una conversazione telefonica mentre è inchiodata a letto per una malattia. Sola in casa, la donna è sempre più terrorizzata.
RaiUno – 0.55

altri modi di organizzare:

Sopra abbiamo visto un archivio organizzato in base al **canale TV**.

A fianco un archivio, organizzato sulla base del **tipo di spettacolo**:

Osserva l'archivio a fianco *"I FILM DEL GIORNO (alla TV sabato 20 sett. 2003)"*. Da quante Unità Informative è formato:

E' un archivio regolare?

Osserva la prima Unità Informativa ed elenca i nomi delle proprietà:

titolo del film

➤ **la progettazione di archivi**

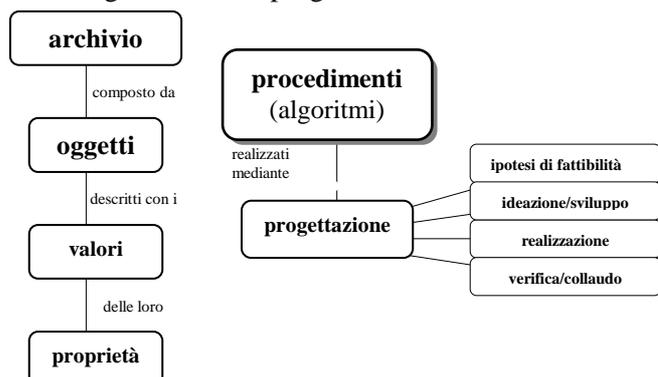
Nella nostra società, che non a caso è chiamata anche "società delle informazioni", le attività di **raccolta**, di **organizzazione/memorizzazione** e di **gestione** di archivi di informazioni sono divenute sempre più rilevanti, soprattutto da quando una parte di esse viene gestita dai calcolatori a velocità elevatissime e a costi sempre più ridotti. Mentre i calcolatori si occupano della gestione, l'attività di raccolta rimane, in buona parte di competenza umana. Essa è fondamentale, visto che in una società caratterizzata dall'abbondanza di informazioni è indispensabile capire quali sono quelle realmente importanti rispetto agli scopi che ci siamo proposti. Per regolamentarla, soprattutto quando essa si occupa di informazioni relative a persone, in tutti i paesi sono state varate apposite **leggi** (675/96 detta anche "legge sulla tutela della privacy"). Esse tendono a impedire che informazioni che il singolo cittadino considera riservate vengano raccolte e divulgate a sua insaputa danneggiando la sua immagine.

L'attività di **progettazione** di un archivio, come ogni attività tecnologica, partirà dall'esame dei bisogni da soddisfare, e si articolerà nelle fasi caratteristiche di ogni attività di progettazione.

Una volta stabilito che l'archivio è **fattibile** si passa alla sua **ideazione**. Definiti quali sono gli **oggetti** che compongono l'archivio, si deciderà di quali delle loro **proprietà** ci interessano i **valori**.

Inoltre, in base al tipo di bisogni da soddisfare, l'archivio potrà essere **aperto** o **chiuso**.

L'**archivio chiuso** è un archivio nel quale i valori delle proprietà sono memorizzati al momento della sua realizzazione.



Ad esempio l'archivio alunni che viene realizzato all'inizio dell'anno scolastico dalla segreteria della scuola e che contiene tutte quelle proprietà (nome più valore) degli alunni necessarie all'attività di segreteria.

L'archivio aperto è un archivio nel quale, al momento della realizzazione, i valori delle proprietà sono quasi del tutto assenti. Verranno inseriti successivamente.

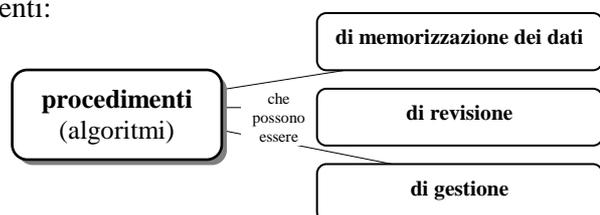
Ad esempio l'archivio alunni del nostro insegnante all'inizio dell'anno scolastico contiene solo il cognome e nome. Si riempirà di tutti gli altri dati riguardanti la valutazione, le assenze, i colloqui con i genitori, nel corso dell'anno scolastico.

E' anche importante stabilire il **formato** (standard) con cui i dati vengono raccolti e memorizzati. Vanno evitati dati destinati a cambiare come l'età, da sostituire con la data di nascita. Anche questo dato va memorizzato in modo da favorirne il controllo automatico (si usano due caratteri per giorno, due per mese, due per anno / ad es: 04-05-92).

La gestione di archivi

Un archivio potrà essere gestito direttamente dall'uomo oppure potrà essere utilizzato il calcolatore. In entrambi i caso andranno preparati i seguenti procedimenti:

Se, ad esempio, la segreteria della scuola progetta un archivio riguardante gli alunni della nostra classe penserà di inserire alcune proprietà come: cognome, nome, data di nascita, via e n° di residenza, comune di residenza, CAP, ecc...

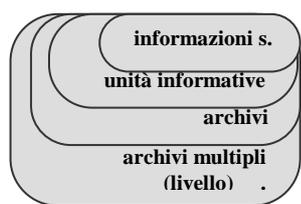


Una volta progettato l'archivio, dovrà innanzitutto predisporre procedimenti di memorizzazione dei dati, che potrà avvenire al momento dell'iscrizione facendo compilare alle **famiglie unità informative** predisposte su appositi moduli. Le unità informative così realizzate andranno a formare l'**archivio** della nostra classe, che a sua volta sarà inserito nell'**archivio multiplo** della nostra scuola.

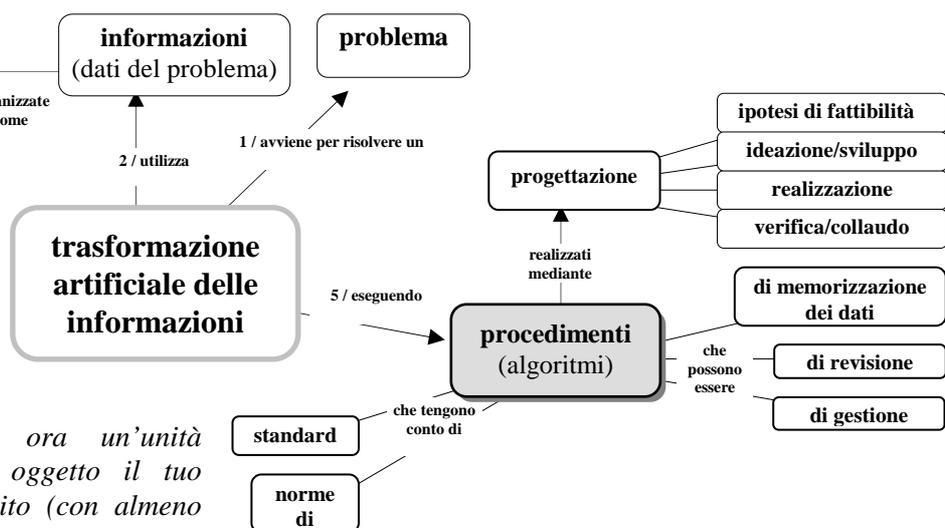
La segreteria dovrà predisporre procedimenti di revisione dei dati nel caso qualche alunno segnali alcune variazioni come il cambio di residenza o di numero di telefono.

Vi saranno infine i procedimenti di gestione, quelli in cui i dati inseriti nell'archivio servono per soddisfare varie richieste: un insegnante per una gita scolastica chiede un elenco degli alunni della classe, un alunno chiede un certificato di iscrizione alla scuola per farsi rilasciare il tesserino per l'autobus, ecc...

Ed ora al lavoro!



La mappa ci ricorda che l'uomo tende a dare un'organizzazione logica ai dati ai dati che raccoglie



Su un foglio realizza ora un'unità informativa che ha per oggetto il tuo giocatore di calcio preferito (con almeno cinque valori di attributi).

Indica poi il nome dell'archivio e dell' archivio multiplo al quale appartiene.

L'ACCESSO AI DATI COLLOCATI SU UN SUPPORTO

Sinora abbiamo visto come e con quali criteri noi raccogliamo ed organizziamo le informazioni. Ora esamineremo come le informazioni, raccolte e depositate su un supporto, possono essere recuperate dall'utente che ne ha bisogno.

Il collocamento di informazioni su un supporto (carta, lavagna, nastro magnetico, ecc...) è un'operazione che noi eseguiamo molto spesso durante la giornata.

esempi: scrivere delle frasi su un foglio / fare un disegno / scattare una fotografia / registrare su un nastro della musica / ecc...

Un'altra operazione che noi compiamo molto spesso è la ricerca di un'informazione precedentemente depositata su un supporto, operazione meno semplice in quanto molto spesso l'informazione che cerchiamo è collocata insieme a molte altre. Normalmente quando si collocano delle informazioni su un supporto si stabilisce già, dal modo con cui le informazioni vengono collocate, quale tipo di **accesso** potrà avere l'utente.

accesso = possibilità da parte dell'utente di raggiungere il dato o i dati desiderati

A seconda di come vengono depositati i dati sul supporto e dal modo con cui poi sarà possibile raggiungerli si hanno diverse **strutture di dati**. In base al tipo di accesso potremo individuare:

- strutture di dati ad accesso sequenziale

si hanno quando l'utente deve esaminare tutte le informazioni ad una ad una sino a trovare l'informazione desiderata.

esempi: trovare una canzone in una cassetta (con il registratore sprovvisto di contagiri) / trovare il volto di un compagno di classe in una fotografia scattata a tutta la classe

in una struttura ad **accesso sequenziale** lo spazio di memoria in cui è collocato il singolo dato si chiama **campo**

- strutture di dati ad accesso diretto

si hanno quando l'utente può arrivare subito all'informazione desiderata, visto che conosce l'**indirizzo** della posizione di memoria (**elemento**) in cui essa è collocata ed è in grado di raggiungerla.

esempi: date delle coordinate e una carta topografica trovare a quale segno convenzionale corrispondono / ascoltare la quarta canzone di un compact disk / leggere il sesto capitolo del proprio libro di scienze / vedere nell'orario settimanale quale materia c'è martedì alla terza ora

in una struttura ad **accesso diretto** lo spazio di memoria in cui è collocato il singolo dato si chiama **elemento**

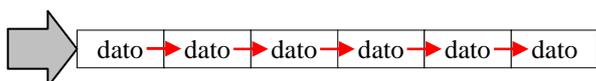
➤ **l'accesso sequenziale**

In queste strutture vi è sempre un dato iniziale, da cui si inizia la lettura, procedendo poi a quelli collocati in successione ad esso. Si legge sempre un dato alla volta. La lettura sequenziale viene spesso agevolata dall'uso di un puntatore (un dito, una matita, ecc) che permette anche di effettuare pause tra la lettura di un dato e quella del dato successivo senza "perdere il segno".

Tra le varie strutture di dati ad accesso sequenziale prendiamo in esame le due più comuni:

• **la lista**

è una sequenza ordinata di dati in cui ciascuno ha un solo predecessore e un solo successore. Ogni dato è accessibile solo dopo un esame ordinato di tutti i dati precedenti.



esempio: se cerchiamo una parola nella pagina di un libro dobbiamo esaminare tutte le parole sino a trovare quella voluta

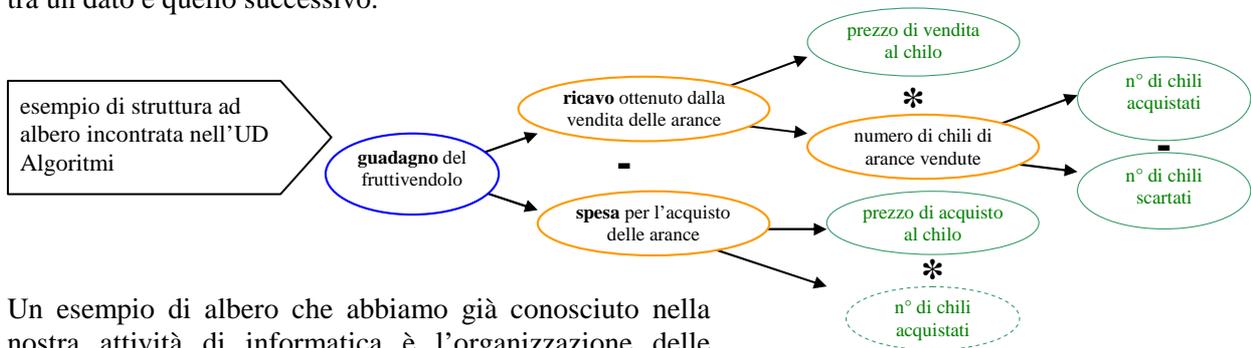
Un esempio di lista che abbiamo già conosciuto nella nostra attività di informatica è il **file**.

Riguardo alle liste bisogna ricordare che:

- 1) la lettura parte sempre dal primo dato della lista
- 2) dopo la lettura di un dato si passa a leggere il dato successivo (a meno che si decida di interrompere)
- 3) leggere non significa mantenere in memoria tutti i dati letti. Se tra 100 dati ne sto cercando solo uno partirò comunque a leggere dal primo dato e così di seguito. Ogni dato sarà sostituito nella memoria da quello seguente sino a quando non avrò trovato quello che cerco.

• **la struttura ad albero**

E' una sequenza di dati che parte da un dato iniziale, chiamato radice, da quale si può accedere a più di un dato. Anche da questi ultimi potranno partire più diramazioni. Di questa struttura, che viene chiamata gerarchica perché mette in rilievo rapporti di dipendenza tra le informazioni precedenti e quelle successive ad esse collegate, abbiamo già visto esempi nei grafi di scomposizione dell'UD Algoritmi. Vengono chiamati nodi i dati che portano ad altri dati mentre i dati terminali sono chiamati foglie. Viene invece chiamato livello di un dato la distanza tra esso e la radice mentre è chiamato ramo il collegamento tra un dato e quello successivo.



Un esempio di albero che abbiamo già conosciuto nella nostra attività di informatica è l'organizzazione delle **cartelle** (directory) di una memoria secondaria hard disk, dischetto, CD, ecc....)

➤ **l'accesso diretto**

Si ha un accesso diretto (o casuale) quando l'utente può arrivare subito all'**elemento** che contiene l'informazione desiderata, visto che ne conosce l'**indirizzo**. Ogni elemento contiene uno o più dati ed è raggiungibile separatamente dagli altri grazie alla chiave d'accesso o indice.

- esempi: date delle coordinate e una carta topografica trovare a quale segno convenzionale corrispondono*
- ascoltare la quarta canzone di un compact disk*
- leggere il sesto capitolo del proprio libro di scienze*
- vedere quale materia c'è martedì alla terza ora*

Tra questo tipo di strutture possiamo individuare due diverse modalità di accesso:

- **mediante chiavi d'accesso**

- **mediante indici**

L'accesso mediante chiavi d'accesso

E' il sistema più diffuso e consiste nel raggiungere il dato grazie ad una o più parole, numeri, segni, ecc... (chiamate **chiavi d'accesso**) al cui significato esso è legato.

- esempi: nell'elenco telefonico i dati relativi ad un certo abbonato sono raggiungibili solo se ne si conosce il cognome*
- nell'orario scolastico si può sapere quale materia c'è ad una certa ora solo conoscendo sia il nome del giorno sia il numero dell'ora*

Proprio perché questa modalità di accesso funziona grazie a legami di significato tra i dati, esso è molto usato negli archivi gestiti direttamente dall'uomo, ma non può essere usato quando a gestire gli archivi è un calcolatore. In questo caso si dovrà utilizzare sempre un accesso diretto mediante indici.

Esaminiamo ora velocemente le due principali strutture di questo tipo:

• **tavola**

Si ha una tavola quando il supporto viene diviso in una serie ordinata di spazi (elementi) raggiungibili separatamente dagli altri, ognuno dei quali viene raggiunto grazie ad un dato iniziale (chiave d'accesso), che permette di raggiungere una serie di dati a cui esso è legato.

esempio: l'elenco telefonico degli apparecchi di una città è un archivio formato da tante unità informative quanti sono gli apparecchi attivi. Dal punto di vista dell'accesso esso è una tavola, infatti solo la conoscenza del valore della proprietà "cognome dell'utente" permette di arrivare ai valori delle altre proprietà della UINF cercata

491	» Sisto, 51 v. XV Aprile	023 542 818	» CHI
494	» PASSEINI Franco, 8 v. XVI Aprile	023 543 380	» CHI
	» RIVA Maria, 2 v. Erida	023 545 406	» CHI
720	» MELZI Angelo, 5 v. Erida	023 545 439	» CHI
721	» MENELLI Alessandra, 34 v. Lac. Sol.	023 545 439	» CHI
723	» MENEGGIO Francesco, 51 v. Balena	023 542 436	» CHI
891	» MENDOZA MIRANDA Rosario Elizabeth	023 542 896	» CHI
892	» M. S. S. S.		» CHI
893	» MENECHHELLA Antonio	023 543 897	» CHI
894	» M. S. S. S.		» CHI
895	» MENECHHELLI Carlo, 32 v. XV Aprile	023 543 872	» CHI
896	» MENECHHELLI Gianmario, 6 v. Garibaldi	023 541 180	» CHI
897	» M. S. S. S.		» CHI
898	» Mengoni, 2 v. Garibaldi	023 541 445	» CHI
899	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
900	» MENGARILLI Giuseppina, 33 v. Costa	023 543 222	» CHI
901	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
902	» MENGHINI Alberto, 5 v. Tonale	023 543 906	» CHI
903	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
904	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
905	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
906	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
907	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
908	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
909	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
910	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
911	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
912	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
913	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
914	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
915	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
916	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
917	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
918	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
919	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI
920	» Mengoni, 2 v. Garibaldi	023 546 141	» CHI

• **tabella a doppia entrata**

Il supporto viene infatti diviso mediante linee orizzontali e verticali formando così spazi (elementi) di grandezza variabile dentro i quali sono collocate le informazioni.

Questa struttura, se viene utilizzata per gestire un archivio, permette un rapido accesso ai valori delle singole proprietà che vengono collocati negli elementi. Infatti ogni fascia orizzontale è destinata ad accogliere nei vari elementi gli attributi dello stesso oggetto, ogni fascia verticale è destinata ad accogliere nei vari elementi lo stesso attributo dei vari oggetti (o viceversa).

esempio: l'orario settimanale della seconda H (nome dell'archivio). E' una tabella a doppia entrata in cui i giorni della settimana sono gli oggetti e le ore di lezione sono i nomi delle proprietà. Per accedere ai valori contenuti nella tabella (le materie insegnate), oltre al suo nome, devo conoscere anche nome del giorno e dell'ora.

L'accesso mediante indici

E' il sistema meno diffuso nella nostra realtà quotidiana, ma è l'unico modo per accedere direttamente ad archivi di dati quando a gestirli sono dei calcolatori.

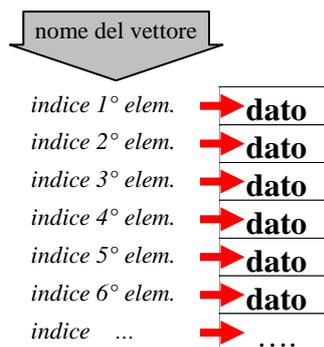
Il dato è raggiungibile grazie ad una o più numeri, caratteri, segni, ecc... (chiamati **indici**) con i quali esso non ha legami di significato e che servono ad individuare lo spazio di memoria in cui esso è depositato.

esempio: in una carta topografica un dato raggiungibile solo conoscendo le coordinate (indici) dello spazio in cui è depositato

Esaminiamo le due principali strutture di questo tipo:

• **vettore** (matrice a una dimensione)

Il supporto viene diviso in una serie ordinata di spazi ognuno dei quali può essere raggiunto grazie ad un indice progressivo. Questo tipo di struttura è poco utilizzato nella realtà quotidiana, infatti numeri od altri segni che noi utilizziamo per accedere ai dati hanno in genere un legame di significato con essi (classifiche, graduatorie, ecc...).



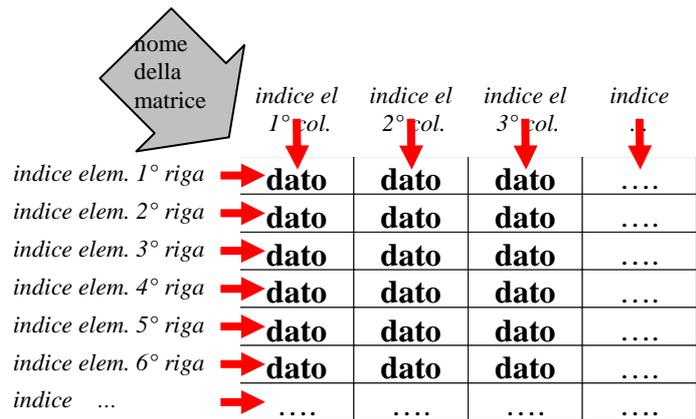
Per ritrovare dei dati in un vettore bisogna avere il nome del vettore e il valore dell'indice dell'elemento cercato.

• **matrice**

Il supporto viene diviso mediante linee orizzontali e verticali. Vengono a formarsi spazi di grandezza variabile dentro le quali sono collocate le informazioni.

Ogni spazio (elemento) può essere raggiunto grazie ad una coppia di indici progressivi, il primo dei quali indica la posizione rispetto alle fasce orizzontali, il secondo quella rispetto alle fasce verticali.

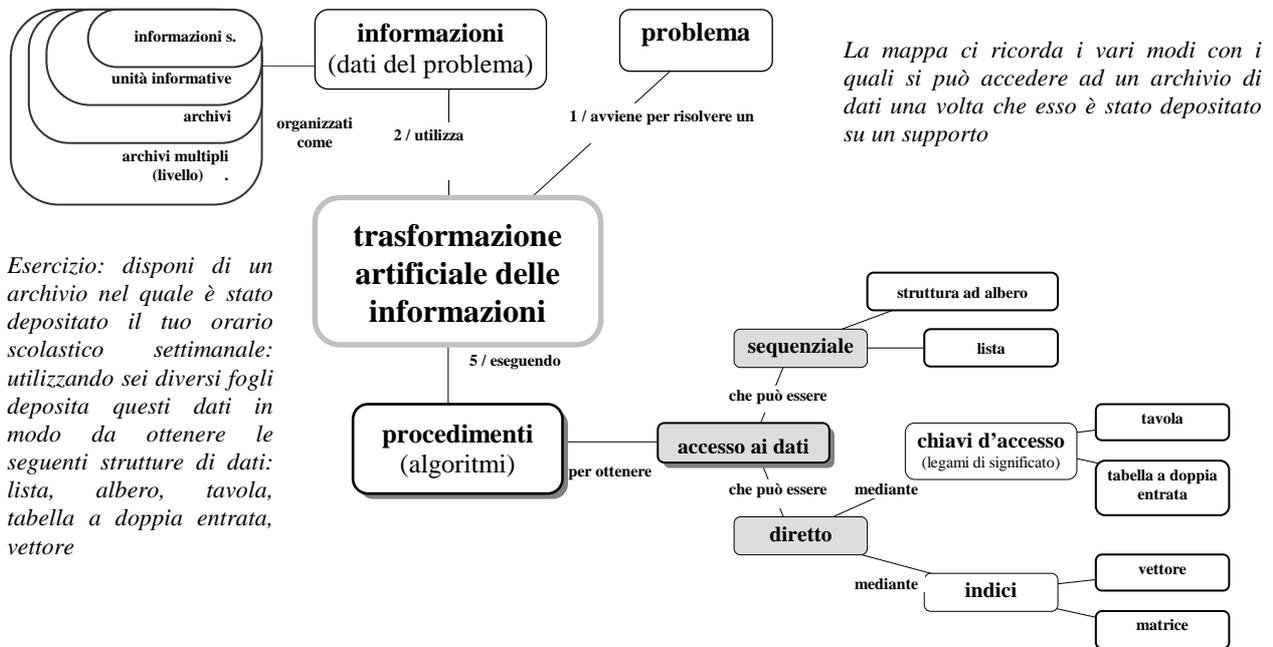
*esempi: una carta topografica è una matrice su cui i dati sono raggiungibili mediante le coordinate
una scacchiera è una matrice; le posizioni dei pezzi sono comunicabili mediante le coordinate del quadrato su cui sono collocati
quando giochiamo a battaglia navale collochiamo le "navi" su una matrice; l'avversario dovrà indovinare le coordinate dei quadratini su cui sono collocate*



Per ritrovare dei dati in una matrice bisogna sapere il nome della matrice e i valori degli indici dell'elemento cercato.

*Esempio: in una carta topografica dell'IGM in scala 1:25000 le coordinate sono indici che permettono di raggiungere direttamente piccolissimi spazi di memoria (quadrati di 0.4*0.4 millimetri sulla carta corrispondenti a 10*10 metri nella realtà)*

Ed ora al lavoro!



STRUTTURE DI DATI NELLA MEMORIA INTERNA DEL CALCOLATORE

Anche nell'area variabili della *memoria centrale* del calcolatore (RAM) è possibile raggruppare sotto un unico nome **strutture di dati** differenti dalla variabile semplice. Queste strutture vengono chiamate variabili complesse o **matrici**. La loro gestione è sostanzialmente diversa da quella che avviene nella nostra mente visto che il calcolatore non è in grado di stabilire legami di significato tra le informazioni che memorizza.

In precedenza abbiamo paragonato le variabili a cassette dentro i quali è possibile memorizzare dei dati, ora diciamo che è possibile raggruppare blocchi di cassette sotto un unico nome che verrà stabilito con le stesse modalità viste per le variabili semplici. Il nome indicherà anche se la struttura è destinata ad ospitare dati numerici o alfanumerici. I cassette appartenenti allo stesso blocco vengono chiamati **elementi** e potranno essere distinti tra di loro grazie a dei numeri progressivi, detti **indici**, che partono sempre dal numero zero.

Se per svolgere un programma è necessario disporre di strutture di dati sarà prima necessario **dimensionarle**, sarà necessario cioè collocare, in genere all'inizio del programma, un'istruzione che, dando nome e caratteristiche, ordini al calcolatore di creare la struttura. Se la parte di RAM che il sistema operativo destina ad ospitare i dati parametrici non è sufficiente, il calcolatore al dimensionamento darà un segnale di errore.

- **il dimensionamento / la dichiarazione**

L'ordine è:

dim nome della struttura (dimensioni della struttura)

Il **dimensionamento** è l'unica operazione che può essere fatta sull'intera struttura. Negli esempi che faremo in seguito, per dimensionare le strutture utilizzeremo dati parametrici. Ad esempio **dim a\$(b)** dove il valore di **b** sarà stato precedentemente chiesto all'operatore.

E' però importante tenere presente che molto moderni linguaggi di programmazione collocano obbligatoriamente il dimensionamento delle strutture all'inizio del programma. Questa fase iniziale, che viene chiamata **dichiarazione**, richiede dei valori costanti che vengono stimati dal programmatore in base alle dimensioni massime previste.

Se ad esempio devo dimensionare una struttura che ospiterà i cognomi degli alunni di una classe di scuola media non supererò il numero di 30 elementi, visto che l'attuale legislazione impedisce che si superi tale numero

- **il ridimensionamento delle strutture**

Con l'ordine:

redim nome della matrice

è possibile ridimensionare nella memoria RAM una o più strutture di dati cancellando i dati esistenti ed, eventualmente, modificandone le dimensioni.

Esaminiamole ora le varie strutture:

➤ **la matrice ad una dimensione o vettore**

E' la struttura ideale quando dobbiamo memorizzare i valori di un'unità informativa. E' formato da una serie di spazi di memoria numerati riuniti sotto un unico nome. Il numero indica la posizione e viene chiamato **indice** mentre lo spazio di memoria si chiama **elemento**. Teoricamente la numerazione parte da 0 ma, in genere si rinuncia ad utilizzare l'elemento 0 e si parte da uno.

Tutte le altre operazioni sui dati presenti nella struttura (assegnazioni, riassegnazioni, slicing ecc..) vanno fatte sui singoli elementi il cui nome è composto dal nome del vettore seguito dalla posizione dell'elemento nella stessa (indice). Essi sono gestibili con le stesse modalità già viste per le semplici variabili.

nome della matrice

indice	elementi
1
2
3
4
..

Mettiamo infatti di voler memorizzare in un vettore tutti i cognomi degli alunni del nostro gruppo di lavoro e poi di scriverli sul monitor. Se avessimo a disposizione solo le semplici variabili dovremmo scrivere un programma molto lungo; tanto più lungo quanto più numerosi sono i dati da confrontare.

E' molto più semplice utilizzare un vettore per depositarvi, nei singoli elementi, i cognomi degli alunni.

Per il dimensionamento si userà l'ordine: **dim nome della matrice (n° righe)**

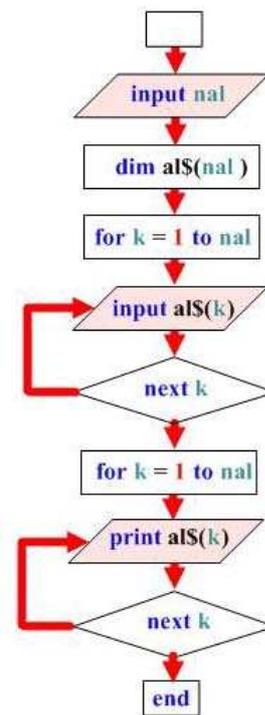
a1\$	indice	elementi
cognome 1°alunno ▶	1
cognome 2°alunno ▶	2
cognome 3°alunno ▶	3
cognome 4°alunno ▶	4
.....

E' possibile dimensionare la matrice con un numero fisso che sicuramente non verrà superato. Visto che nel nostro corso i gruppi non superano mai i cinque alunni con : **dim a1\$(5)** saremo sicuri di poter contenere tutti i dati da inserire. Gli elementi eventualmente in eccesso rimarranno vuoti. Comunque nel nostro programma scegliamo di dimensionare in base al numero di alunni precedentemente chiesto.

Vediamo ora il programma da usare per risolvere il problema dato. A fianco vi è il diagramma di flusso nel quale vanno inserite le istruzioni in BASIC. Nel diagramma di flusso vengono messe le istruzioni essenziali tralasciando la parte relativa alla comunicazione che invece va inserita nel programma.

' immissione dei dati

```
input "Scrivi il numero di alunni del gruppo "; nal
dim a1$(nal)
for k=1 to nal
    print "alunno "; k :input "Cognome dell'alunno "; a1$(k)
next k
' scrittura su monitor
cls:print "I componenti del gruppo sono:"
for k=1 to nal
    print k, a1$(k)
next k
end
```



Come abbiamo detto su ogni elemento sono possibili tutte le operazioni già viste per le singole variabili.

Se ad esempio inserisco al termine del programma appena visto: **print mid\$(a1\$(2),1,3)** il calcolatore scriverà i primi tre caratteri del cognome del secondo componente del gruppo.

Per la gestione degli indici, come abbiamo visto nell'esempio, è fondamentale l'uso dei cicli.

La presenza di un unico nome per strutture di dati che possono contenere anche centinaia di informazioni e la possibilità di gestire degli indici parametrici con dei cicli rende molto più semplici i programmi di gestione.

➤ **la matrice bidimensionale**

Se invece di memorizzare solo i cognomi dei componenti del nostro gruppo, decidiamo di memorizzare, per ogni componente del gruppo, altre proprietà avrò bisogno di una struttura diversa. Infatti per ogni componente del gruppo dovrò formare un'unità informativa. Insieme formeranno un archivio.

Vediamo dunque cosa succederà se, oltre al cognome, voglio memorizzare anche il nome e l'indirizzo.

Per il dimensionamento si userà l'ordine:

dim nome della matrice (n° righe , n° colonne)

Dimensioneremo dunque una matrice con un numero di righe equivalente al numero di alunni che compongono il gruppo (che andrà richiesto) e tre colonne, una per ognuna delle proprietà che ho deciso di memorizzare.

Nel programma che segue organizziamo l'immissione e la scrittura dei dati con un ciclo nidificato.

Il ciclo esterno serve a contare gli alunni e dunque le righe della matrice. Il ciclo interno serve a contare le proprietà e dunque le colonne della matrice.

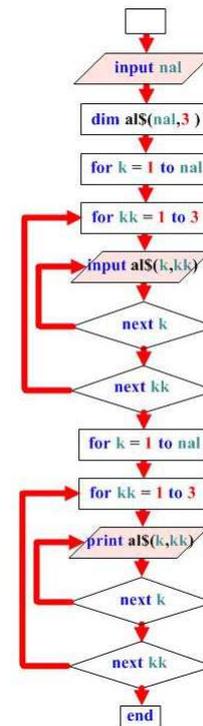
' immissione dei dati

```
input "Scrivi il numero di alunni del gruppo "; nal
dim al$(nal, 3)
for k=1 to nal
  for kk=1 to 3
    print "alunno "; k , "proprietà numero "; kk
    input "scrivi il dato da memorizzare "; al$(k, kk)
  next kk:cls
next k
' scrittura su monitor
cls:print "dati dei componenti del gruppo:"
for k=1 to nal
  for kk=1 to 3
    print k, al$(k, kk)
  next kk
next k
end
```

Il programma appena scritto funziona bene ma è poco chiaro dal punto di vista della comunicazione, specie nell'immissione dei dati. Per migliorarlo possiamo rinunciare al ciclo delle proprietà, chiedendone i singoli valori. Il programma (a fianco) sarà più lungo ma più chiaro.

Potresti provare, con lo stesso sistema, o con altri, a migliorare anche la scrittura su monitor.

al\$	indice	cognome	nome	indirizzo
1°alunno ▶	1
2°alunno ▶	2
3°alunno ▶	3
4°alunno ▶	4
.....



' immissione dei dati

```
input "Scrivi il numero di alunni del gruppo "; nal
dim al$(nal, 3)
for k=1 to nal
  print "alunno n°"; k , "scrivi il cognome ";
  input al$(k, 1)
  print "alunno n°"; k , "scrivi il nome ";
  input al$(k, 2)
  print "alunno n°"; k , "scrivi l'indirizzo ";
  input al$(k, 3)
next k
' scrittura su monitor
cls:print "dati dei componenti del gruppo:"
for k=1 to nal
  for kk=1 to 3
    print k, al$(k, kk)
  next kk
next k
end
```

Per rendere più semplice il nostro lavoro, i programmi realizzati riguardavano piccole matrici. Molto spesso si lavora con matrici più capienti. Il nostro insegnante, ad esempio, gestisce i dati riguardanti gli alunni della nostra classe con una matrice dimensionata con 30 righe (n° massimo di alunni) e 10 colonne.

> **la matrice tridimensionale**

Immaginiamo di avere l'esigenza di memorizzare le tre proprietà già citate (cognome, nome e indirizzo) anche per gli alunni degli altri sei gruppi di lavoro della nostra classe. In questo caso è utile utilizzare una matrice tridimensionale. Essa può essere utile anche per memorizzare, in un'unica struttura, le tre classi del nostro corso. Potrò dimensionare una matrice tridimensionale con l'ordine:

dim nome della matrice (n°fogli **, n°righe **, n°colonne ***)

- *da far corrispondere al n° dei gruppi
- **da far corrispondere al n° degli alunni
- ***da far corrispondere al n° delle proprietà

La matrice tridimensionale è dunque la somma di più matrici bidimensionali (che vengono chiamate fogli) riunite sotto un unico nome. Essa viene utilizzata per gestire facilmente con il calcolatore un archivio multiplo (vedi la fase 2).

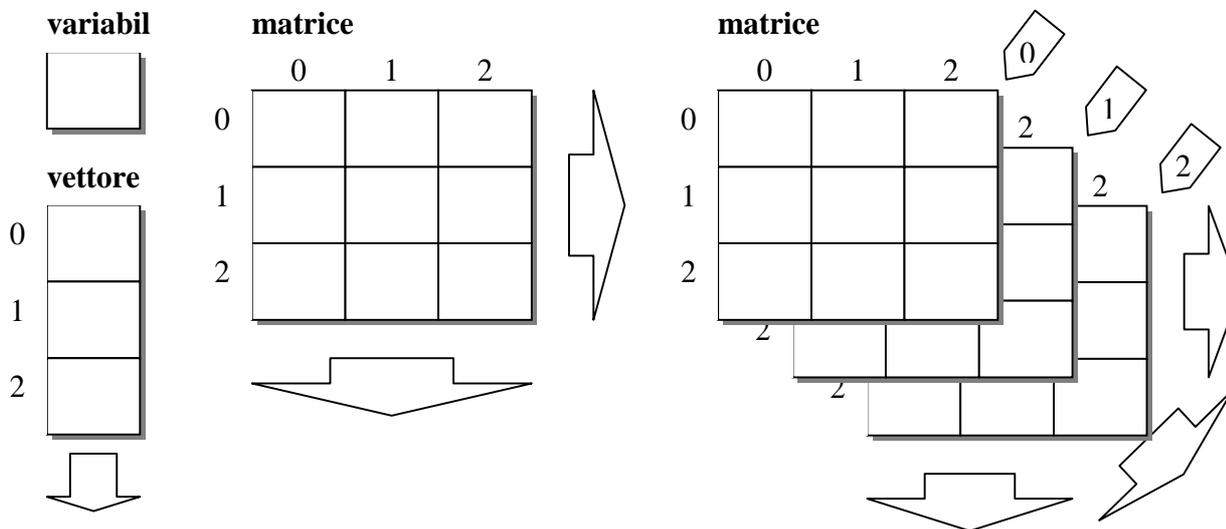
In genere vengono utilizzati tre cicli (uno per i fogli, uno per le righe, uno per le colonne).

Visto che JUSTBASIC non è in grado di gestire matrici tridimensionali, rimandiamo le attività su questa struttura all'anno prossimo.

al\$

1
2
3	1
4	2	1
..	3	2
	4	3
	..	4
	

I seguenti disegni illustrano le caratteristiche delle principali strutture organizzabili nella memoria RAM del calcolatore:

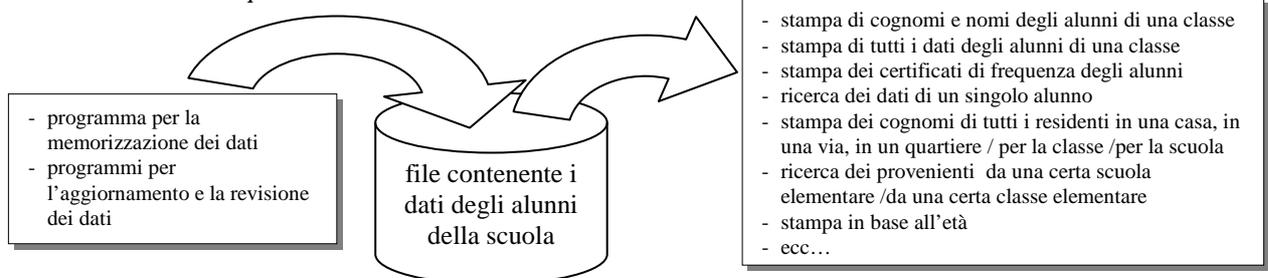


Anche in questo caso fondamentale è la **regolarità**: tutti i fogli devono infatti avere lo stesso numero di righe e di colonne. Questo significa che il numero di righe da usare nel dimensionamento corrisponderà al numero di unità informative dell'archivio più numeroso. Le righe degli altri fogli, in mancanza di dati, resteranno vuote.

INGRESSO E USCITA VERSO LE MEMORIE SECONDARIE

Come sappiamo (vedi fase 3 dell’U.di A. Algoritmi) un file di dati è rappresentato da una lunga fila di campi separati tra di loro dal **segno di fine campo**. Quando il programmatore organizza il trasferimento di un archivio di dati su un file ne dovrà progettare con cura l’organizzazione. Il sistema migliore è quello di farne un disegno su un foglio di carta, disegno che sarà utile quando dovranno essere realizzati i programmi di gestione che prevederanno la lettura dei dati contenuti in quel file. Infatti, una volta che i dati di un archivio sono stati depositati su un file, possono essere numerose le applicazioni che ne possono prevedere l’utilizzo.

Vediamone un esempio:



Nella progettazione del file:

- chiameremo **campo** lo spazio di memoria raggiungibile sequenzialmente (in questo caso esso ospita un singolo valore)
- chiameremo **record** la somma degli spazi di memoria (campi) che saranno occupati dai valori di un’unità informativa.

E’ bene tenere presente che mentre il campo rappresenta un’entità realmente esistente sul file, il record è un’entità logica che serve al programmatore per gestire in modo ottimale i dati contenuti nel file.

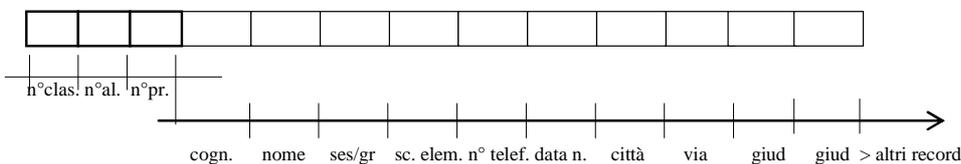
E’ anche importante, quanto si progetta un archivio, essere previdenti ed eventualmente raccogliere anche quelle informazioni che possono sembrare non immediatamente utili.

Proprio perché un file depositato su disco deve essere disponibile per diverse utilizzazioni è bene che esso contenga nella sua parte iniziale tutti quei dati che saranno necessari per poterne effettuare la lettura. Un file contenente dei dati sugli alunni di una classe difficilmente potrà essere letto se al suo inizio non sono registrati il numero di alunni della classe (numero di **oggetti**) e il numero di dati registrati per ogni alunno (numero di **proprietà**). I campi iniziali contenenti i dati necessari per la lettura dei record che formano un file vengono chiamati **campi di testa**.

Vediamo dunque di realizzare un programma che legge dall’hard disk i dati degli alunni della nostra classe (file 2 H realizzato con Blocco Note) e li memorizza in una matrice bidimensionale.

sotto mappa che illustra il modo con cui i dati sono organizzati nel file 2 H.txt e di fianco come appare lo stesso file aperto con Blocco note:

campi di testa > record contenenti i dati degli alunni

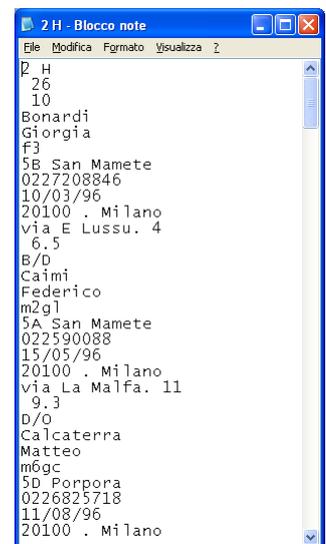


mappa che illustra come viene organizzata la RAM del calcolatore per accogliere i dati provenienti dall’hard disk:

- cl\$
- nal
- npr
- k
- kk

matrice a1\$

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
...										



cl\$ m

Il programma carica i campi di testa e poi i dati degli alunni dal file 2 H.txt (nell’apertura manca il percorso e dunque il file deve essere collocato nella stessa cartella da cui è stato caricato il programma) e li memorizza nelle strutture organizzate in RAM.

```
open"2 H.txt" for input as#1
input #1, cI$, nal, nat
dim al$(nal, nat)
for k=1 to nal
for kk=1 to nat
input #1, al$(k,kk)
next kk
next k
close #1
end
```

Se vogliamo verificare il suo corretto funzionamento possiamo collocare prima della fine l’ordine **print al\$(3,8)**. Il calcolatore scriverà l’indirizzo (colonna 8) del terzo alunno (riga 3) della classe.

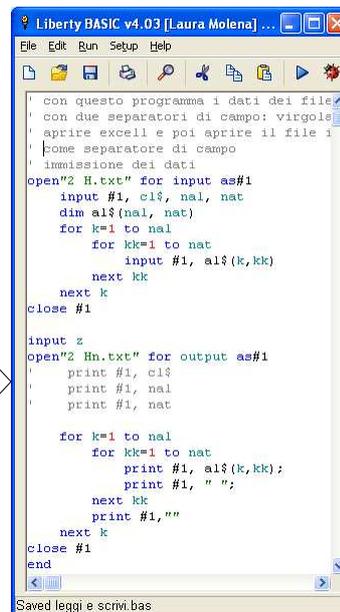
➤ **l’importazione dei dati (da Blocco note ad Excel)**

Nella gestione degli archivi di dati, il lavoro più faticoso è quello di archiviare nuovi dati. E’ un’attività che, per ora, resta a carico dell’uomo, anche se la ricerca anche in questo campo ci sta preparando grosse novità. Se i dati dell’archivio sono stati già stati memorizzati su una tabella di Word, di Excel o di Access ci sarà facile trasferirli da un’applicazione all’altra.

Se i dati sono memorizzati su Blocco Note in modo sequenziale (come il file 2 H.txt), vanno modificati. Dovendo inserire i dati del file in una tabella, come prima cosa dovremo togliere i dati che non fanno parte dell’archivio, cioè i campi di testa che occupano le prime tre righe del file. Nel file tutti i dati della stessa unità informativa vanno portati sulla stessa riga separati da una virgola. Il segno di “a capo” rimane per separare le diverse Unità informative (e dunque le righe della tabella) mentre la virgola servirà a separare le colonne.

il file “2 Hn.txt” modificato in modo da poter essere inserito in foglio di Excel. L’inserimento delle virgole al posto degli “a capo” può essere fatto manualmente o realizzando un apposito programma.

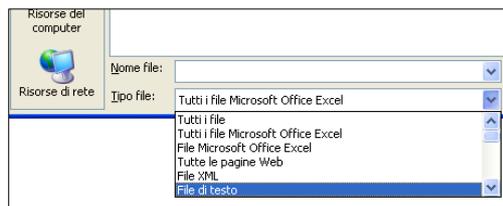
programma realizzato per leggere il file “2 H.txt” e risalarlo modificando il formato dei dati.



A questo punto il file in Blocco note può essere portato su un foglio di Excel.

Si apre Excel e si clicca su **apri**. Nella finestra che si apre selezionare come **Tipo di file** la voce **file di testo**.

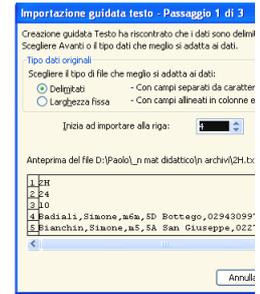
Si apre ora la finestra di importazione guidata.



Alla richiesta **Inizia ad importare dalla riga** selezioneremo dunque **4**.
 Nella nuova finestra ci appaiono ora solo i dati degli alunni.



Comunichiamo ora al calcolatore di considerare come segno di delimitazione le virgole (oltre al segno di “a capo”) spuntando l’apposita voce. Nella finestra il calcolatore al posto delle virgole inserirà le colonne.



Cliccando **Avanti** e poi **Fine** il calcolatore conclude l’importazione guidata e mostra il foglio di Excel con i dati appena importati.

➤ **Il segnale di “fine del file” (EOF)**

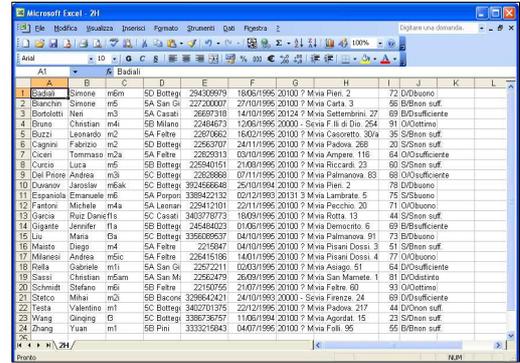
Quando viene eseguito l’ordine di chiusura il drive registra in coda al file che si sta chiudendo un apposito segnale che viene chiamato **segno di fine file** (End Of File).

In BASIC vi è un apposito ordine studiato per seguire la lettura di un file. Esso è molto utile nel caso non sia possibile conoscere il numero di campi che formano il file. In questo caso l’individuazione del segno di fine file può evitare di proseguire la lettura del file con il conseguente segnale di errore.

L’ordine: **eof (n° canale)**

conterrà il valore 0 se dopo il campo appena letto vi è il segno di fine campo mentre conterrà il valore -1 se dopo la lettura del campo è stato rilevato il segno di fine del file. In questo caso vuol dire che si è arrivati alla fine del file e che dunque è il momento di chiuderlo.

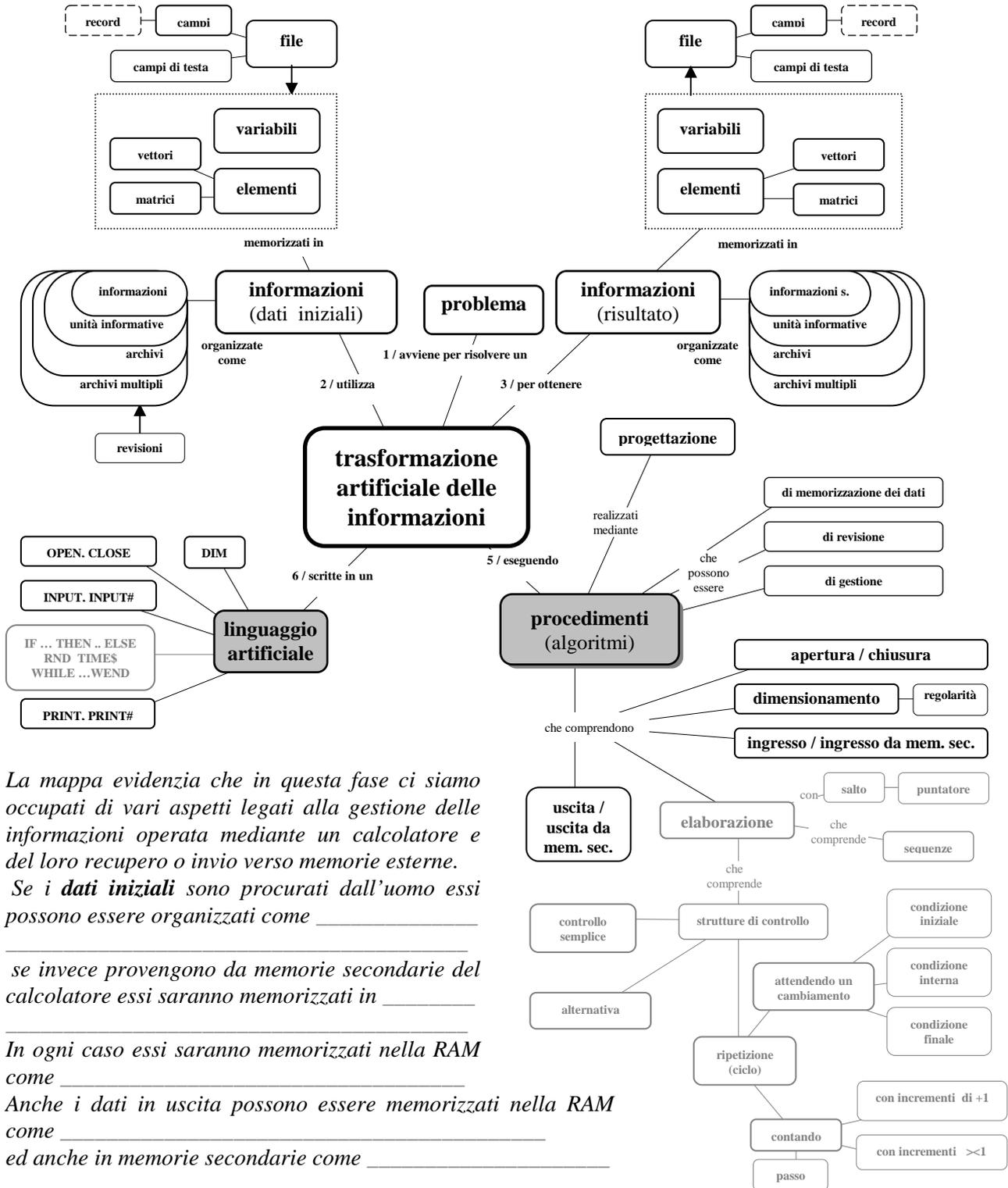
Ad esempio il programma a fianco ci permetterà di leggere un file chiamato **2H.txt** che registra i dati di un numero imprecisato di alunni. Con questo programma, però, i dati una volta letti vanno persi. Cosa devo fare se invece li voglio mantenere nella RAM del calcolatore?



```

open "2 H.txt" for input as#1
do while eof(#1) = 0
input #1, a$
print a$
loop
close #1
print "fine"
end
    
```

Ed ora al lavoro!



La mappa evidenzia che in questa fase ci siamo occupati di vari aspetti legati alla gestione delle informazioni operata mediante un calcolatore e del loro recupero o invio verso memorie esterne.
 Se i **dati iniziali** sono procurati dall'uomo essi possono essere organizzati come _____

se invece provengono da memorie secondarie del calcolatore essi saranno memorizzati in _____

In ogni caso essi saranno memorizzati nella RAM come _____

Anche i dati in uscita possono essere memorizzati nella RAM come _____
 ed anche in memorie secondarie come _____

Ci siamo occupati degli aspetti procedurali e in particolare del **dimensionamento** che permette di _____

e delle istruzioni che consentono tale operazione come _____

E dell'**apertura e chiusura** che permette di _____
 utilizzando gli ordini _____

Le strutture di dati archiviate nel calcolatore devono essere **regolari**; significa che _____

ARCHIVIARE I DATI CON I DATA BASE

Per chi non è in grado di gestire archivi di dati realizzando in proprio del software di gestione esiste un'applicazione appositamente realizzata. Abbiamo già studiato sul libro le caratteristiche principali di un data base realizzato con Access, ora vediamo come importare in Access l'archivio di dati della nostra classe e come realizzare su di esso delle ricerche di dati.

➤ preparazione del data base

Come prima cosa va aperto il data base. Access chiede subito di salvare e dunque bisogna scegliere una cartella dove depositare i propri lavori ed anche un nome con cui salvare il progetto.

Nella finestra che ci viene presentata selezioneremo **Crea una tabella in visualizzazione struttura**. Ora ci viene richiesto di scrivere i nomi dei campi che formano la tabella.

Ricordiamo che in Access vengono chiamati **campi** le **colonne** che formano una tabella; le **righe** sono chiamate **record**.

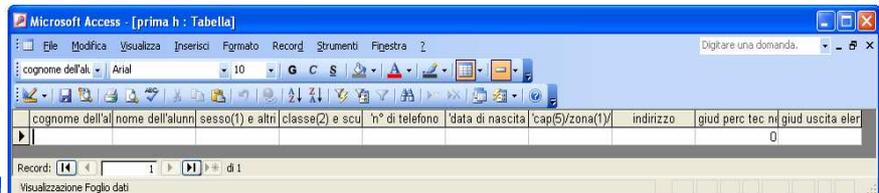
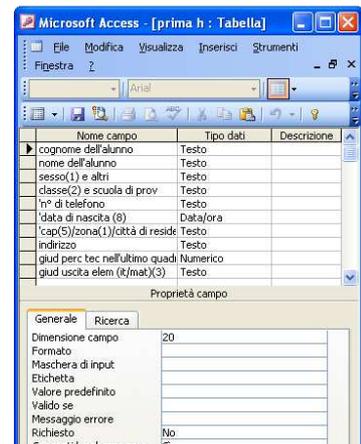
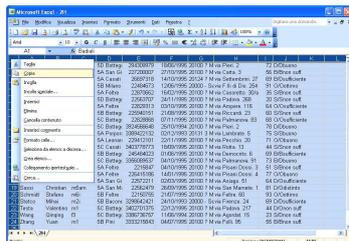
Visto che conosciamo già l'archivio che intendiamo inserire nella tabella, non avremo difficoltà a scrivere i nomi dei dieci campi e a definire la tipologia dei dati che verranno inseriti. . Tranne la data e il giudizio di Tecnologia, gli altri dati possono essere lasciati nella tipologia **testo**. E' anche bene portare da 50 a 20 il numero dei caratteri richiesti.

Abbiamo dunque preparato la struttura della nostra tabella che ora possiamo salvare. La salveremo con il nome **seconda H** rifiutando di stabilire una chiave primaria (non essendoci altre tabelle, non serve). Passiamo ora ad occuparci dei dati che dovremo importare nella tabella appena preparata.

➤ l'importazione dei dati (da Excel ad Access)

Ora possiamo concludere la prima parte del nostro lavoro portando i dati dal foglio di Excel, organizzato nella fase precedente, alla tabella di Access appena preparata. L'importare i dati direttamente da Blocco Note è teoricamente possibile ma più complicato e dunque lo evitiamo.

La riapriamo e passiamo subito in modalità **visualizzazione Foglio dati**.

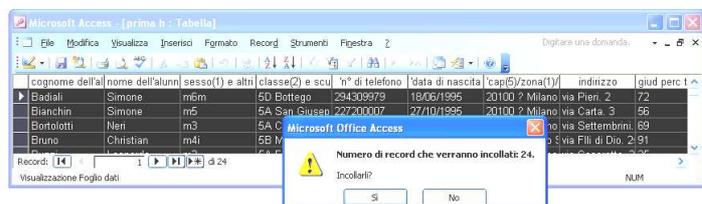
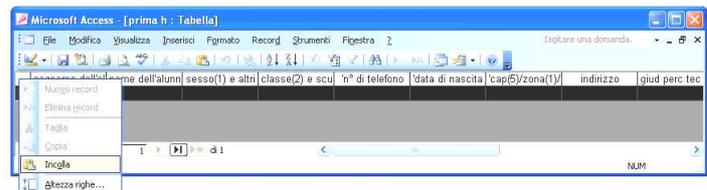


Torniamo sul foglio di Excel e clicchiamo sul quadratino in alto a sinistra (origine delle righe e delle colonne). Tutti i dati del foglio verranno selezionati e noi cliccheremo su copia (con il tasto destro o sulla barra degli strumenti).

Ora anche su Access selezioniamo il quadratino in alto a sinistra e clicchiamo su **Incolla**.

Il calcolatore ci chiede conferma dell'ordine segnalandoci quanti record saranno incollati.

Dopo la conferma ci apparirà la tabella **seconda H** oramai completa.



In modalità **visualizzazione Foglio dati** è anche possibile con **File > carica dati esterni > importa**. Nella finestra di importazione che si apre selezionare come Tipo di file > Microsoft Excel

➤ **L'importanza della regolarità**

Il procedimento è simile anche se i dati sono su una tabella di Word. Basta evidenziare la tabella e copiarla per poi riportarla in Access con lo stesso procedimento.

L'importante è ricordare, quando si fanno queste operazioni, che gli **archivi copiati devono essere regolari**. Tutto ciò che non c'entra con l'archivio (intestazioni comprese) va eliminato prima di copiare. Potremo ora salvare e prepararci alla seconda parte del nostro lavoro: la ricerca di dati.

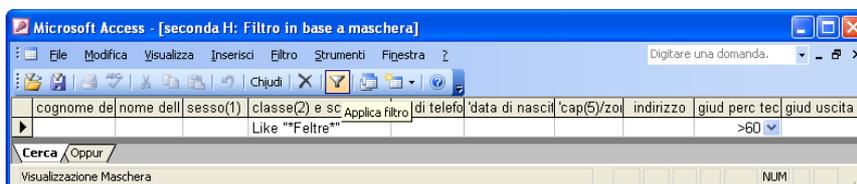
➤ **consultazione del data base (la maschera)**

Vediamo ora come cercare nella tabella **seconda H** i record contenenti dati che ci interessano operando una consultazione dell'archivio mediante filtro. Facciamo un esempio:

- o desidero cercare tutti gli alunni che vengono dalla scuola elementare Feltre e che abbiano almeno la sufficienza in Tecnologia

Dopo aver aperto la tabella in *visualizzazione Foglio dati*, posso utilizzare **record > filtro > filtro in base a maschera** oppure cliccare sull'icona  nella barra degli strumenti.

Compare la **maschera di ricerca** dove compaiono i nomi dei campi. Nel campo **classe(2)** e **scuola di prov** viene scritto ***Feltre*** preceduta e seguita da un asterisco.



Gli asterischi segnalano al calcolatore che cerchiamo la parola **Feltre** anche preceduta e seguita da altri caratteri. Nel campo **giud perc tec nell'ultimo quadrimestre**, che è numerico, scriviamo **>60**. Il calcolatore cercherà tutti i giudizi maggiori di 60.

Possiamo notare che il calcolatore, appena si lascia la cella di scrittura, contorna la parola cercata con delle virgolette se essa è inserita in un campo di testo. Nel caso vi siano asterischi il calcolatore fa precedere la parola da **Like**, che indica una ricerca su una parte del contenuto del campo.

Selezioniamo poi **filtro > applica filtro/ordina** oppure clicchiamo su .



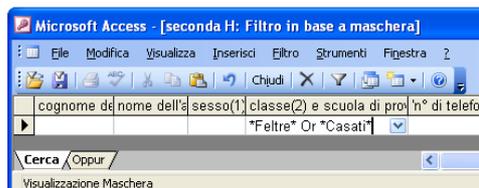
Il calcolatore effettuerà la ricerca e mostrerà i record della tabella che contengono i dati cercati.

Cliccando nuovamente su  ricomparirà nuovamente la tabella di origine.

➤ **utilizzo dei connettivi per cercare con la maschera**

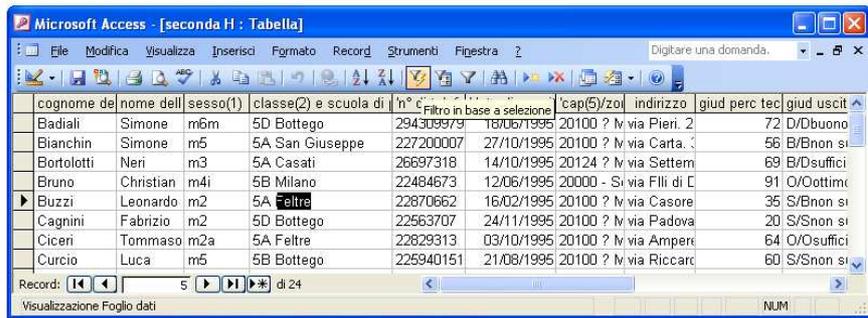
La **maschera di ricerca** permette di aggiungere ancora altri criteri di ricerca e permette anche di utilizzare i connettivi **And**, **Or** e **Not**. Mettiamo ad esempio di voler cercare i record degli alunni che vengono dalla scuola elementare Feltre oppure dalla scuola elementare Casati e che abbiano almeno la sufficienza in Tecnologia.

Selezionata la maschera scriveremo ***Feltre* Or *Casati*** (ricordarsi di separare l'ordine **Or** dai caratteri che precedono e che seguono). L'istruzione che verrà subito riconosciuta dal calcolatore. Applicato il filtro, otterremo tutti i record degli alunni che provengono da una delle due scuole cercate e che hanno la sufficienza.



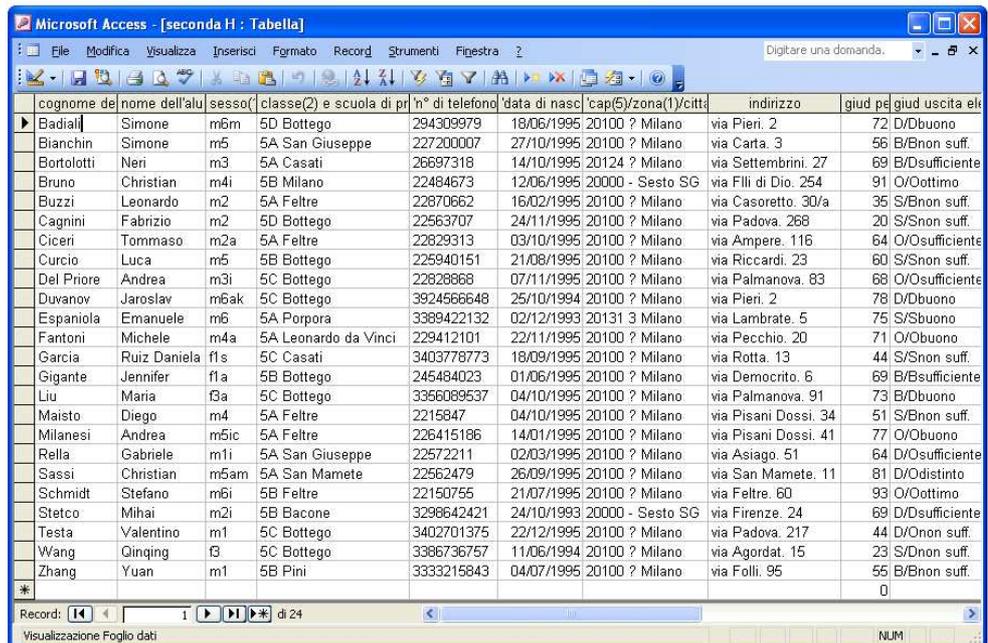
➤ **consultazione del data base (la selezione)**

Se invece dobbiamo cercare una parola sola, possiamo selezionarla direttamente sulla tabella. Poi si clicca sull'icona che indica il **filtro in base a selezione** .



ed ora al lavoro!

Osserva la tabella a fianco e rispondi alle seguenti domande:



Scrivi in dettaglio (una per una) tutte le azioni che devi fare per ottenere i record degli alunni maschi che abitano in via Pisani Dossi

Scrivi in dettaglio (una per una) tutte le azioni che devi fare per ottenere i record degli alunni nati nei mesi di maggio(05), giugno(06) o luglio(07)
