

CELLA DI MEMORIA:	CONTENUTO:	OPERAZIONE:
...

Problematiche di controllo dei dati

Non vi è dubbio che uno dei problemi più importanti in ogni attività di gestione dati sia quella del loro controllo, che consiste o in un processo di validazione iniziale per cui i dati sono «filtrati» nel momento dell'immissione, oppure in una attività di controllo a posteriori, per mezzo della quale i dati «scorretti» vengono evidenziati e/o eliminati.

Le forme di controllo sono abbastanza varie, si va da un controllo di formato (ad esempio su un campo DATA non si può immettere il 29/02/85), a un controllo di esistenza (ad esempio un campo SIGLA DI PROVINCIA non deve accettare una sigla non esistente) oppure controllo logico, ad esempio di tipo «incrociato», così detto in quanto mette in relazione più campi. Ad esempio se c'è un campo DATA DI INIZIO e uno DATA DI FINE è necessario che la prima sia anteriore alla seconda.

È evidente l'importanza della funzione di controllo, infatti non esiste nessun archivio o database che possa ammettere, per definizione, dati sbagliati. Pensate all'archivio dell'Anagrafe del Comune di Roma, oppure all'Archivio del Magazzino dei pezzi di ricambio Fiat.

Si può dire che buona parte dell'attività dei programmatori «tradizionali» sia proprio quella di costruire routine e programmi di controllo dei dati, sia in procedure di acquisizione dei dati che in procedure di elaborazione degli stessi.

Addirittura, ai primordi dell'informatica, quando l'unico modo per comunicare i dati al calcolatore era la perforazione delle schede, il sistema più economico per il controllo dei dati in immissione era costruito dalla doppia digitazione, chiamata verifica.

Con la nascita dell'informatica individuale e con la diffusione dei nuovi prodotti software «user friendly» il concetto di controllo ha subito delle trasformazioni. L'utente individuale, che usa per un suo problema specifico un computer e un software, immettendo e gestendo in prima persona i dati di suo interesse, può non aver bisogno di eseguire i controlli.

Ad esempio per costruire un archivio dischi nel campo velocità metterà o 33 o 45. Se poi sbaglia e immette 35 o 43 o qualsiasi altra cosa non è grave.

Se invece il prodotto è professionale e chi lavora è un programmatore, la faccenda cambia aspetto, e lo cambia per vari motivi.

Il primo motivo è che la persona che immette i dati non è mai la stessa che li usa e quindi è meno interessata a che i dati siano esatti, la seconda è che questa persona deve immettere molti dati e quindi sul grande numero aumentano le probabilità di errore, la terza è che l'uso che si fa di questi dati è importante e quindi non sono ammessi errori, che possono avere conseguenze gravi.

Il quarto motivo, specificamente tecnico, è quello che è sempre opportuno, quando si può, codificare i dati. In quanto il campo codificato è molto più facile sia da controllare al momento della immissione che da elaborare successivamente (ad esempio A=33 giri B=45 giri C=CD).

Vediamo come queste semplici considerazioni si trasportano in una tematica tipo spreadsheet, che, nascendo come simulatore... di foglio di carta a quadretti, permette un uso spregiudicato ed estemporaneo. Ciò non vuol dire che però, quando occorre, in applicazioni importanti, lo spreadsheet non possa permettere sofisticate e efficaci funzioni di controllo e validazione dati.

Quello che vogliamo esaminare è proprio questa problematica, indipendentemente però da un ambito applicativo specifico. Vedremo come realizzare controlli di campi numerici, di stringa, di campi data e controlli incrociati.

Come appena detto tratteremo controlli spiccioli, eseguiti con formule che occupano poche caselle, anzi preferibilmente una sola casella.

Questo sia perché in definitiva questa rubrica raccoglie suggerimenti, trucchi del mestiere, e non applicazioni, neanche se di limitate dimensioni, sia perché la sperimentazione suggerita sia facilmente eseguita da chiunque, impiegando pochi minuti, che diventano ben spesi se poi il suggerimento trova pratica applicazione.

Premessa agli esempi

Prima di passare ad esaminare gli esempi, realizzati tutti con il SIMPHONY in versione italiana, occorre fare alcune precisazioni, riguardanti il modo di presentare gli esempi stessi.

Nello spreadsheet, al contrario di un qualsiasi linguaggio, ad esempio Basic, non è possibile eseguire listati, attraverso i quali comprendere la meccanica delle operazioni logico-matematiche, e eseguire output che evidenzino i risultati relativi.

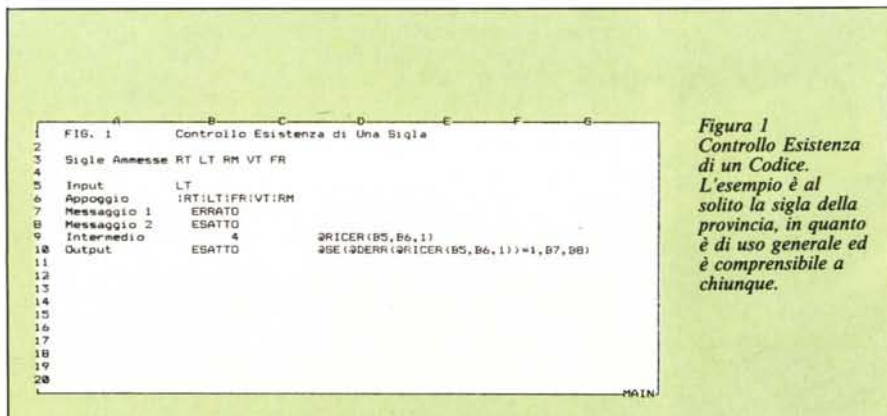


Figura 1
Controllo Esistenza di un Codice.
L'esempio è al solito la sigla della provincia, in quanto è di uso generale ed è comprensibile a chiunque.

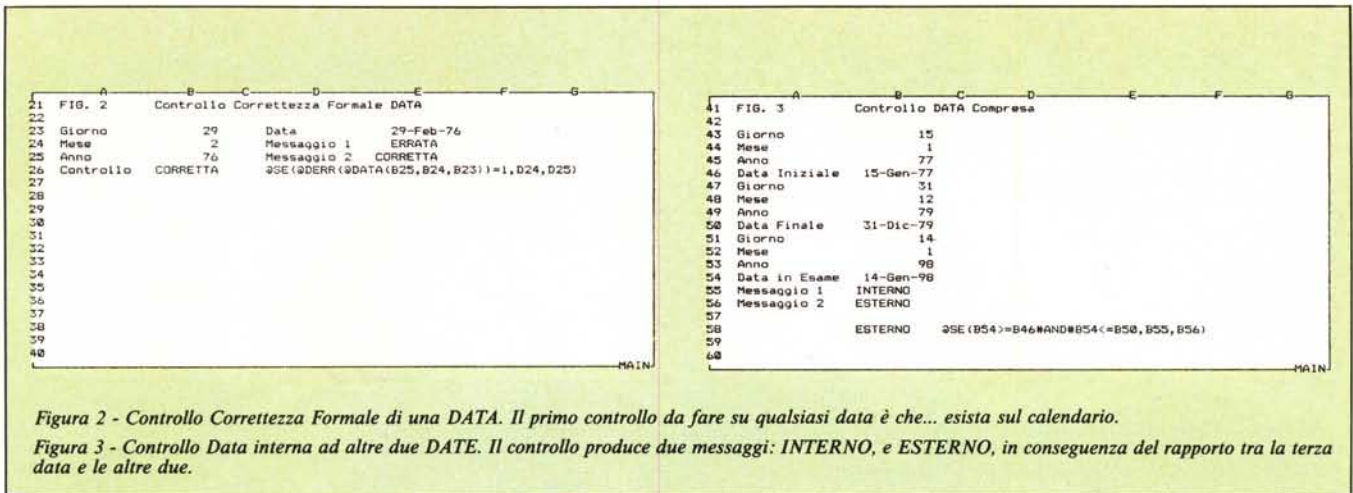


Figura 2 - Controllo Correttezza Formale di una DATA. Il primo controllo da fare su qualsiasi data è che... esista sul calendario.

Figura 3 - Controllo Data interna ad altre due DATE. Il controllo produce due messaggi: INTERNO, e ESTERNO, in conseguenza del rapporto tra la terza data e le altre due.

Il tabellone ha due livelli: quello esteriore facilmente riproducibile, ma che non dice assolutamente nulla di cosa, ovvero quali formule ci siano sotto. C'è poi, appunto, il livello sottostante, che a sua volta non ha senso se non legato rigidamente al contenuto delle celle cui si riferisce.

Inoltre un uso avanzato del tabellone comporta il fatto di concentrare in una sola casella, e quindi in una sola formula, la soluzione di un certo problema con il risultato che la formula sarà sì solo una, ma sarà tanto complicata.

È quindi estremamente difficile presentare degli esempi in cui siano chiaramente individuabili i due livelli di contenuti.

Il compromesso che abbiamo raggiunto consiste in due scelte. La prima è che riproduciamo, sotto forma di stringa, all'interno dell'esempio stesso la formula che risolve il relativo problema.

La seconda scelta consiste nello snocciolare le formule troppo lunghe in più parti, legate a catena, l'una all'altra, in modo da capire i passi logici della soluzione. In questa operazione

di traduzione gestiamo anche la messaggistica via caselle.

Trasportando la soluzione suggerita in un ambito applicativo specifico sarà opportuno accorpate le formule in una sola, che al limite comprenda anche la messaggistica, tenendo presente che il SIMPHONY, con il quale realizziamo gli esempi, permette formule lunghe 240 caratteri.

Controllo di Esistenza di una Sigla - Figura 1

Uno dei controlli più diffusi è quello di un codice che deve essere presente in un certo elenco.

Se tale codice è di pochi caratteri e il numero dei codici possibili è sufficientemente corto è possibile risolvere il problema in una sola casella utilizzando una funzione di Ricerca di Sottostringa (@ RICER).

Se il codice è di più caratteri, bisogna prevenire il pericolo che venga riconosciuto anche un codice composto da due porzioni di codici contigui. Ad

esempio se sono accettati i codici AA e BB e la stringa campione è «AABB», le sottostringhe di due caratteri riconosciute sono AA BB ma anche AB.

Per evitare tale evenienza suggeriamo di inserire un carattere non presente sulla tastiera, o particolarmente raro e difficile da digitare, all'interno della stringa campione, ad esempio il carattere ASCII 092, per isolare i vari codici l'uno dall'altro, per cui nell'esempio citato la stringa campione diventa «AA\$BB».

Se la funzione di ricerca di sottostringa non trova il codice, si verifica una condizione di errore che può essere testata con la funzione @ DERR, che restituisce il valore 1 se si è verificato un errore, oppure 0.

Per gestire la messaggistica ERRATO / ESATTO si ricorre alla funzione @ SE, che verifica lo stato di questo FLAG.

In pratica nella casella di input va immessa la sigla, e nella casella di output appare il messaggio ESATTO / ERRATO a seconda che il codice sia presente nella stringa campione o meno.

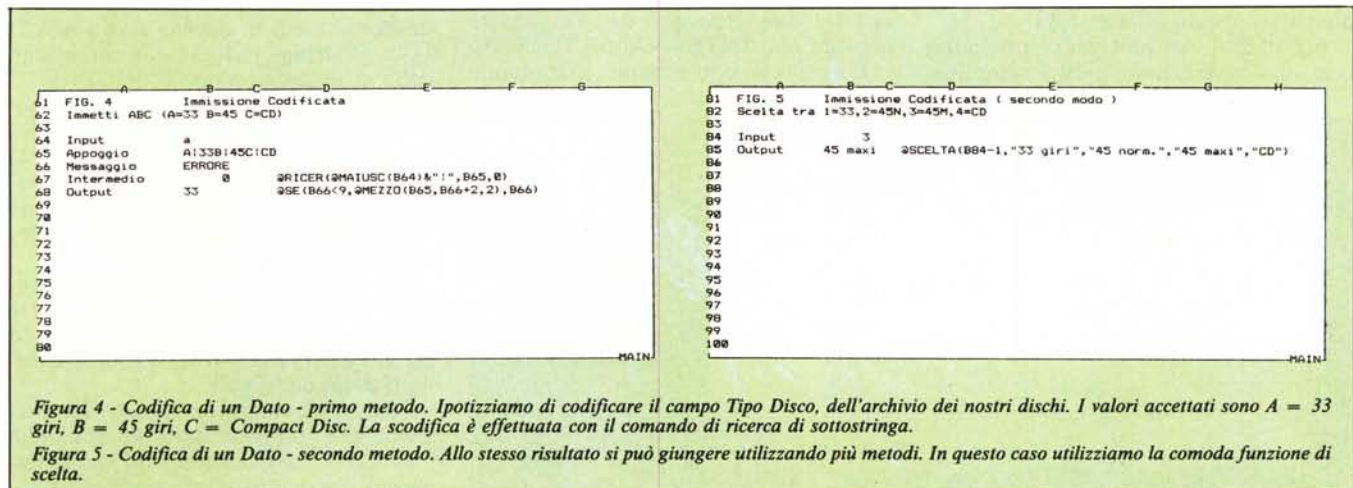


Figura 4 - Codifica di un Dato - primo metodo. Ipotizziamo di codificare il campo Tipo Disco, dell'archivio dei nostri dischi. I valori accettati sono A = 33 giri, B = 45 giri, C = Compact Disc. La scodifica è effettuata con il comando di ricerca di sottostringa.

Figura 5 - Codifica di un Dato - secondo metodo. Allo stesso risultato si può giungere utilizzando più metodi. In questo caso utilizziamo la comoda funzione di scelta.

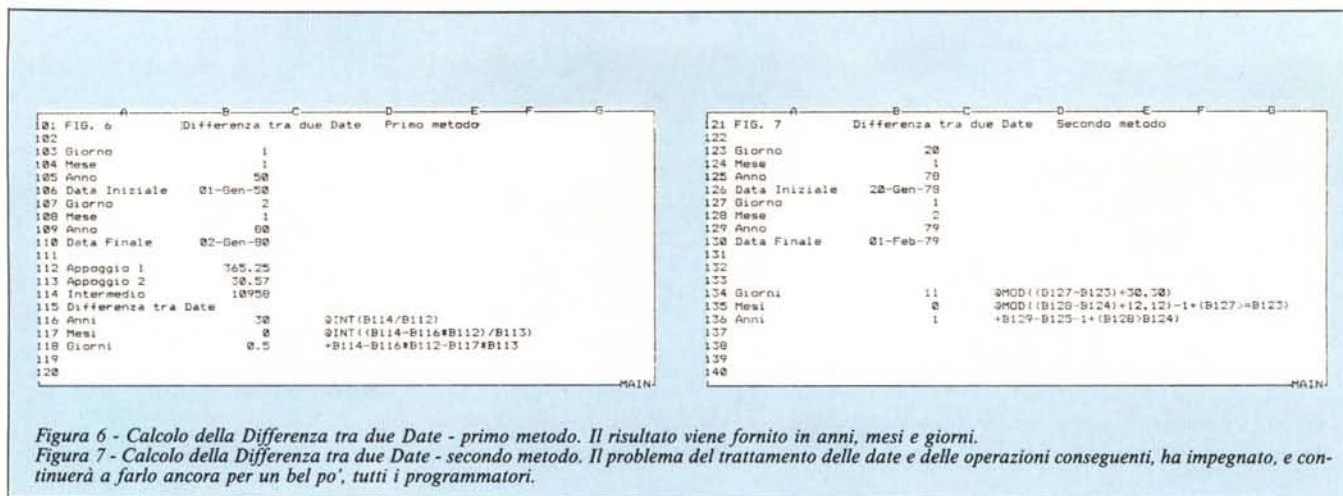


Figura 6 - Calcolo della Differenza tra due Date - primo metodo. Il risultato viene fornito in anni, mesi e giorni.

Figura 7 - Calcolo della Differenza tra due Date - secondo metodo. Il problema del trattamento delle date e delle operazioni conseguenti, ha impegnato, e continuerà a farlo ancora per un bel po', tutti i programmatori.

Controllo Correttezza

Formato Data - Figura 2

Il campo data ha sempre costituito, e costituirà, una spina nel fianco dei programmatori, in quanto pur essendo la data un elemento numerico, non segue non solo nessuna progressione numerica, ma neanche nessuna logica numerica. Sull'argomento sono apparsi recentemente sulla nostra rivista una serie di articoli.

Questo ha comportato l'ulteriore complicazione che in taluni ambiti applicativi (leggi BANCHE) si adottano metodi di calcolo semplificati (mesi di 30 giorni), ma che presentano gravosi problemi di traduzione.

Gli spreadsheet delle ultime generazioni dispongono di numerose funzioni data, con le quali da una data è possibile risalire al giorno, mese, anno e viceversa, oppure ad un numero intero progressivo che rappresenta un numero di giorni sul quale intervenire con operazioni di somma e/o differenza.

Tutto questo agevola la soluzione di parecchi problemi, ma non tutti, e per questi occorre ancora inventare degli algoritmi specifici di calcolo.

Scendiamo nel pratico, cominciamo con una operazione molto semplice,

quella che si rende necessaria per controllare la validità formale di una data.

In tre celle diverse immettiamo tre valori numerici indicanti giorno, mese e anno. Tali tre valori numerici sono inseriti in una funzione @ DATA (gg, mm, aa) che li mette in forma di data.

È evidente che se i tre numeri non sono coerenti con una DATA la funzione darà ERRORE. Con la funzione @ DERR, che fornisce un valore 1 o 0 a seconda della presenza o meno di un errore, si testa la condizione anomala nella casella e si gestisce la messaggistica.

Controllo Data Compresa

Figura 3

Si tratta di un controllo doppio, in quanto la terza data deve essere compresa tra le prime due.

Poiché è possibile eseguire confronti tra date, ottenendo come risultati valori logici 1 o 0, è anche possibile inserire tale confronto in una funzione @ SE, nella quale è anche possibile inserire i due messaggi da visualizzare nei due casi INTERNA e ESTERNA.

È evidente, e lo si vede chiaramente

nella formula, che le due condizioni sono legate da un AND logico, per cui per ottenere la risposta INTERNA, debbono essere ambedue soddisfatte.

Immissione codificata

due metodi - Figure 4 e 5

Questo esercizio è simile al primo, con la differenza che questa volta desideriamo anche la scodifica del codice immesso.

Vogliamo inoltre mostrare come, con uno strumento versatile come lo spreadsheet, sia in generale possibile trovare molte soluzioni, e tutte accettabili anche dal punto di vista «economia», ad uno stesso problema.

L'esempio è relativo ad una codifica della tipologia disco, da utilizzare nel nostro archivio dischi. Ipotizziamo di avere dischi a 33 giri (codice A), a 45 (B) e Compact Disc (C).

Nella casella input inseriamo il codice. Poi sottoponiamo il codice stesso ad una serie di trattamenti. Lo traduciamo in maiuscolo con la funzione @ MAIUSCOL, lo colleghiamo al carattere «sconosciuto» (composizione di stringhe &), in questo caso l'ASCII 124 e la stringa risultante la cerchiamo nella stringa campione (@ RICER).

Se la stringa campione contiene codici e scodifica, è possibile, elaborando il valore numerico ottenuto con la funzione RICER, eseguire una successiva estrazione con la funzione @ MEZZO (stringa, car. iniz., num. car.).

Con la funzione @ SE possiamo anche testare la correttezza del codice e gestire una messaggistica di errore, in questo caso alternativa al codice ritornato dalla codifica.

Molto più semplice è il secondo metodo, che però utilizza la funzione @ SCELTA, che accetta solo valori numerici interi e progressivi, che inoltre

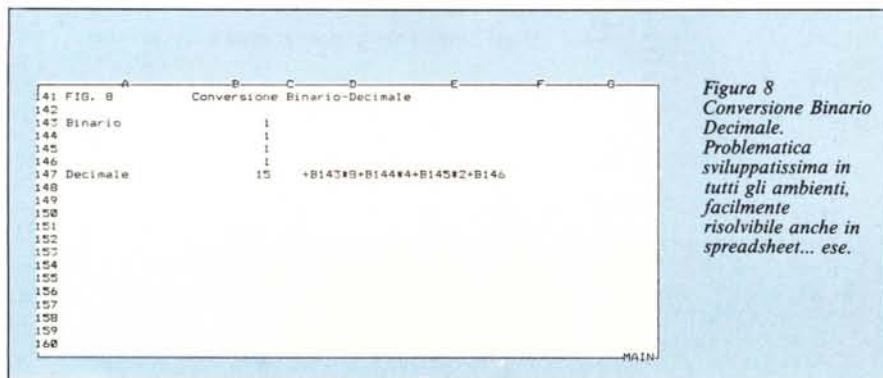


Figura 8 Conversione Binario Decimale. Problematica sviluppatissima in tutti gli ambienti, facilmente risolvibile anche in spreadsheet... ese.

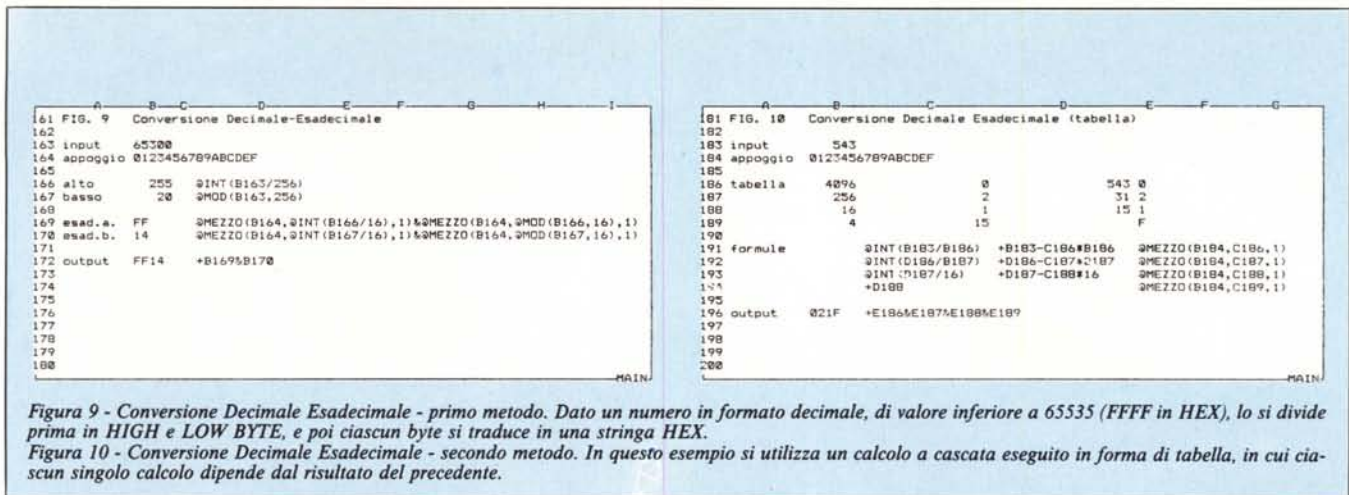


Figura 9 - Conversione Decimale Esadecimale - primo metodo. Dato un numero in formato decimale, di valore inferiore a 65535 (FFFF in HEX), lo si divide prima in HIGH e LOW BYTE, e poi ciascun byte si traduce in una stringa HEX.

Figura 10 - Conversione Decimale Esadecimale - secondo metodo. In questo esempio si utilizza un calcolo a cascata eseguito in forma di tabella, in cui ciascun singolo calcolo dipende dal risultato del precedente.

partono da 0. Quindi se la vostra codifica può essere fatta con codici numerici 1,2,3 ecc. la scelta @SCELTA è la scelta più conveniente (bisticcio voluto).

Tra l'altro se si immette un codice errato la risposta della funzione è il messaggio di errore standard, che è proprio quello che serve.

Differenza tra due Date due metodi - Figure 6 e 7

La differenza tra due date può essere espressa in giorni, oppure in anni, mesi e giorni. Nel primo caso non esistono problemi di calcolo in quanto il risultato è univoco. Nel secondo caso occorre decidere con quale metodo eseguire il calcolo, se in maniera convenzionale o in maniera effettiva.

Ad esempio tra il 15-02 e il 15-03 possono passare 28, 29 oppure un mese, a seconda che si contino i giorni di calendario e l'anno sia o meno bisestile, oppure che si conti il giorno all'interno del mese.

Se esiste una funzionalità per mezzo della quale ciascuna data viene tradotta in un numero intero progressivo di giorni trascorsi da una data iniziale (in SIMPHONY dal 01-01-1900), la differenza tra le date fornisce un numero di giorni trascorsi.

Per tradurre tale numero in anni, mesi, giorni si può ricorrere a metodi convenzionali, ad esempio, con anno e mese di lunghezza in giorni convenzionale (nel nostro esempio di fig. 6 365.25 gg e 30.57 gg). In tal modo però non si riesce a governare completamente il risultato, specie per periodi particolari o per periodi molto lunghi, in quanto come risultato delle varie operazioni eseguite si hanno anche valori decimali.

Il secondo metodo (fig. 7) invece

non ricorre alle funzioni di data ma esegue calcoli separati di giorni, di mesi e di anni, e opera le necessarie correzioni e riporti. Ad esempio tra il 20-01 e il 10-02 la differenza di giorni è di -10 ma va corretta in 20 e il riporto negativo va sottratto alla differenza di mesi che matematicamente invece darebbe 1.

Questo metodo si usa nel calcolo dei ratei di interesse in quanto lavora su mesi di 30 giorni, anzi il valore giorno 31 andrebbe sempre corretto in 30. Anche tale metodo è convenzionale in quanto scompaiono alcuni giorni dell'anno, ma note le convenzioni, produce dati univoci.

Conversione Binario Decimale

Figura 8

Altra tematica tipica, sempre palestra di programmatori alle prime esperienze, è quella della conversione tra le numerazioni decimali, binari e esadecimali.

La più semplice è la conversione tra binaria a decimale, in quanto necessita solo di operazioni matematiche, legate alla posizione dello 0 o 1 nel numero base. Ad esempio il quartultimo elemento del numero di base rappresenta il numero 8, per cui va moltiplicato per 8, il successivo per 4 e così via. I risultati di tutte le moltiplicazioni vengono sommati e si ottiene il corrispondente numero decimale.

Conversioni Decimali Esadecimali due metodi - Figure 9 e 10

Le cose si complicano passando alla conversione da decimale in esadecimale in quanto occorre comunque eseguire operazioni di stringa.

Nel primo esercizio (fig. 9) partiamo da un numero intero (al massimo 65535) e lo traduciamo nei due byte alto e basso. Ognuna di queste due parti a sua volta viene divisa in due dividendola per 16, oppure prendendo il resto della stessa divisione.

In tal modo si ottengono due valori compresi tra 0 e 15 (e cioè tra 0 e F) in HEX. È ora facile, con la funzione @MEZZO eseguita sulla stringa 0123456789ABCDEF ricavare il corrispondente valore esadecimale, da accorpate agli altri.

Vogliamo far notare come le formule siano state diluite in una serie di 5 formule. Teoricamente si possono comporre in un'unica formula, che avrebbe il vantaggio di «risiedere» in una casella ma lo svantaggio di non essere visualizzabile tutta insieme in una videata. Un altro metodo è mostrato in figura 10 e si avvale di una tabellina in cui vengono eseguiti calcoli a caduta. Il numero viene diviso dapprima per 4096, la parte intera del risultato viene scodificata nel carattere HEX corrispondente, mentre il resto continua la sua caduta nella tabella.

Il resto viene diviso per 256 e la parte intera del risultato diventa il secondo valore HEX, e così via.

Un risultato più diretto si ottiene utilizzando la funzione MOD, il cui significato è simulato con la formula scritta nella seconda colonna.

Non descriviamo ulteriormente l'esercizio in quanto gli interessati eseguendolo lo capiranno completamente.

In conclusione vogliamo far di nuovo notare la grande versatilità dello strumento capace non solo di adeguarsi agli ambiti applicativi più svariati, ma anche di permettere soluzioni alternative ad uno stesso problema, altrettanto valide, la cui scelta dipende dal... gusto e dalle attitudini dell'utente.

LA POTENZA DELLO SCSI !

dalla memoria di massa
al controllo di processo



- memorie di massa magnetiche interne ed esterne.
- memorie di massa ottiche.
- unità di back-up
- espansione di memoria RAM

XEBEC
The Zero Defect Company

XEBEC
System

Lungotevere Flaminio, 66
Tel. 06/399544 - 399920
Telefax 3960503 - Tex 623140
00196 ROMA - ITALIA

Ho letto la Vostra pubblicità e vorrei altre informazioni su:

.....
Nome

Società Tel.....

Via

CAP Città