

Numeri Casuali

Generazione ed utilizzo

di Francesco Petroni

Chi comincia ad usare un computer e un prodotto software per fare calcoli, ad esempio un foglio elettronico, vede funzionalità di cui capisce subito l'importanza, e quindi di cui impara subito il funzionamento (si pensi alla funzione che, in un foglio elettronico, serve per sommare una colonna di numeri) e vede funzionalità di cui non capisce subito l'importanza, che tende quindi ad ignorare, oppure vede funzionalità che, in altri casi, lo lasciano interdetto

Una di queste funzionalità, o meglio una di queste funzioni, è quella che serve a generare dei numeri casuali. Mi è capitato spesso di dover spiegare a persone, abbastanza inesperte, la funzione che, in un foglio elettronico, genera dei numeri casuali e di avere una reazione di incredulità. Anzi quando spiegavo che la funzione genera un numero a «casaccio» (a Roma si dice così) compreso tra 0 e 1,1 escluso, quasi sempre mi hanno guardato come se li volessi prendere in giro.

La funzione che genera dei numeri casuali è una funzione molto comoda in numerosi casi applicativi, e addirittura indispensabile in altri. Al punto che certe volte, se si usano prodotti che non dispongono di tali funzioni, occorre in qualche maniera simularla.

Finalità dell'articolo è duplice, vedere come funziona, e se non disponibile, come creare una funzione che genera un numero casuale, e poi vedere alcuni casi di utilizzo di tale funzione, sia casi dimostrativi, ad esempio per verificare che la funzione generi dei numeri casuali effettivamente casuali, sia casi applicativi in cui usare la funzione per i nostri scopi più o meno nobili. La useremo per creare automaticamente migliaia di dati che ci serviranno, ad esempio, per eseguire una «prova di carico» di un'applicazione gestionale, la useremo per estrarre i numeri della Tombola e così via.

Come funziona un «motorino» di generazione di numeri casuali

In un computer la casualità in assoluto non può esistere. Le routine di generazione dei numeri casuali si basano quindi su processi di calcolo molto complessi, anzi talmente complessi, che è impossibile prevedere il risultato, e

```
* generazione di dati casuale (un testo, una data, un numero)
clear
k=1                                && contatore record
w=1                                && contatore righe a video
s=.4334                            && seme iniziale
do while k<22                       && ne genera 22
  t=val(subs(time(),7,2))
  r=t^s-int(t^s)
  s=r
* parola con un numero di caratteri da 5 a 14
  l=int(r*9)+5
  i=1
  c=""
  do while i<=l
    r=t^s-int(t^s)
    s=r
    c=c+chr(int(r*26)+65)
    i=i+1
  enddo
* data tra il 1/1/1930 e il 31/12/94
  r=t^s-int(t^s)
  d=ctod("01/01/30")+int(r*23700)
  s=r
* numero tra 4.000.000 e 12.000.000 arrotondato a 1.000
  r=t^s-int(t^s)
  n=int((r*8000)+4000)*1000
  s=r
*
  @ w,10 say c
  @ w,30 say d
  @ w,50 say n picture "###,###,###"
  w=w+1
* per scaricare su archivio
*   APPEND BLANK
*   REPLACE TESTO WITH C,DATA WITH D,NUMERO WITH N
* altro record
  k=k+1
enddo
```

Figura 1 - dBase - Generazione Automatica di un testo, una data e un numero. I casi sono due, o il prodotto che si utilizza dispone di una Funzione che si occupa di generare un numero casuale, oppure, se non ne dispone, occorre crearla in proprio. Il linguaggio dBase ricade in questa seconda categoria di prodotti. Nella figura il listato di un programma che genera un Campo Testuale, un Campo Data e un Campo Numerico.

quindi tale risultato, se effettivamente imprevedibile, diventa un numero casuale.

Ad esempio se io chiedessi qual è la parte decimale del numero ottenuto elevando a 15,546 il numero 1,2095 nessuno lo saprebbe dire. Proviamo con la calcolatrice, nella parte scientifica, di Windows:

15,546 (X^Y) 1,2095 =
27,62248671322

La parte decimale è quindi 0,62248671322, e, in quanto imprevedibile, è un numero casuale.

Prendiamo poi questo numero, aggiungiamogli 1, e usiamolo come esponente del nostro numero iniziale, quindi:

15,546 (X^Y) 1,62248671322 =
85,77988280191

Ed otteniamo il nuovo casuale 0,77988280191, altrettanto imprevedibile, e così via.

In pratica:

– la generazione di un numero casuale compreso tra 0 e 1 (1 escluso), si ottiene creando un algoritmo di calcolo così complicato che nessuno riesce a prevederne il risultato;

– l'imprevedibilità sta non tanto nel tipo di operazione (ad esempio noi abbiamo usato un'elevazione a potenza) quanto nel fatto che viene presa solo la parte decimale. Inoltre la parte decimale può essere compresa tra 0, se il numero ottenuto è un numero intero, e 1, 1 escluso, perché se fosse 1 sarebbe il numero intero successivo;

– il numero di partenza si chiama Seme (il nostro seme è 15,546). Se il seme è lo stesso e l'algoritmo è sempre lo stesso la serie di numeri casuali sarà sempre la stessa. Cosa che può far comodo in applicazioni di simulazione statistica, ma è del tutto inaccettabile in un'applicazione di tipo gioco. Si pensi ai numeri della Roulette.

Quindi se dovessimo realizzare una routine di generazione di numeri casuali in un prodotto che non dispone della funzione ad hoc, potremmo inventarci noi un algoritmo, in cui un qualsiasi risultato entra nel calcolo del numero successivo.

Se volessimo evitare serie sempre uguali dovremmo generare, in un modo sufficientemente imprevedibile, anche il primo seme, e magari anche i successivi. In generale si utilizza la funzione TIME, presente in qualsiasi prodotto, dalla quale potremmo ad esempio estrarre il numero dei secondi.

I prodotti che dispongono della fun-

Dbase		
PNIEWIZID	07/01/69	6.683.000
NQZJLAA	19/04/37	7.894.000
JUADOCIEWU	08/11/76	10.078.000
ODRI DREVOM	02/07/38	8.696.000
FFDOZII DRCJSN	01/10/51	6.084.000
ZDONPKSUQ	08/05/42	11.650.000
RTGKXT	23/12/64	9.534.000
ACKADONQMLX	16/03/66	6.158.000
TXXYYZEXNC	29/03/52	7.061.000
JOPSI BIACMSUC	06/05/62	11.068.000
PWNJMS	16/10/30	4.353.000
UDPJUX	10/11/69	11.054.000
QBPZNUBGPYXZ	23/03/68	4.961.000
RNBFMMUS	10/11/35	7.026.000
TXKUGNEURMNT	22/06/78	5.904.000
MI BEWKBF	05/11/38	9.038.000
ICLYRLFGL	26/08/42	4.088.000
ETWUWU	29/06/69	10.839.000
HUKGOYOC	05/05/58	10.269.000
EULSDPIHYZ	02/07/70	6.893.000
LRUACMKEY	13/12/52	8.415.000

Figura 2 - dBase - Generazione Automatica di un testo, una data e un numero - Output. Così appare l'output a video del programma mostrato nella figura precedente. Nel testo si parla di «prova di carico» di un'applicazione dBase. In Ingegneria Civile per prova di carico di un ponte si intende il fatto che si riempie il ponte di camion a pieno carico, posti uno vicino agli altri, e si vede... se crolla. Se non crolla (non crolla mai perché le strutture sono dimensionate calcolando X volte il pieno carico) il ponte funziona. Stesso discorso per un'applicazione software, se deve lavorare con 100.000 record occorre eseguire immediatamente una prova di carico a tabelle piene.

zione di generazione del numero casuale incorporano tali algoritmi, per cui, per l'utente, è sufficiente richiamare la funzione per avere il numero casuale. Alcuni, ad esempio il Basic, in tutte le sue varianti, dispone anche del comando Randomize (mentre la funzione è RND) che serve proprio a far variare la serie.

Come addomesticare un numero casuale

È improbabile che un numero casuale serva così come è, e cioè compreso tra 0 e 1,1 escluso. Se lo si prende e lo si moltiplica per 6, si otterrà un numero compreso tra 0 e 6,6 escluso. Se gli si aggiunge 1, tra 1 e 7,7 escluso. Se se ne prende la parte intera, da 1 a 6. E abbiamo simulato il lancio di un dado.

Con lo stesso sistema potremmo generare un numero intero tra 1.000.000 e 2.000.000, magari arrotondato «alle 1.000 lire», oppure un numero intero tra 65 e 91, che corrisponde ai caratteri, nella codifica ASCII, che vanno da A a Z. E così abbiamo usato il generatore di numeri casuali per generare un carattere alfabetico casuale.

Ora cominciamo a vedere qualche applicazione di questi semplici strumenti, e rimandiamo a dopo l'affrontare un argomento un po' più spinoso,

che è la non ripetizione del numero estratto.

Come generare, in mancanza di una specifica funzione, un numero casuale

Cominciamo vedendo un programma scritto in linguaggio xBase.

Non costruiamo una Function, cosa che con le istruzioni dBase III più classiche non è possibile, ma riscriviamo gli algoritmi ogni volta che ci serve, anche più volte nello stesso listato (che vediamo in fig. 1).

A cosa può servire un programma del genere? Può servire ad alimentare in maniera automatica, generando quindi un certo numero di record (1.000, 5.000, 10.000 o quanti se ne vogliono) e caricandoli di dati differenti l'un record dagli altri. Può servire quindi quando si voglia eseguire una «prova di carico» di un'applicazione appena sviluppata, provandola ad archivi pieni.

Eseguiamo un ciclo da 1 a 10.000 (nel nostro listato è fino a 21), all'interno del quale costruiamo tre variabili. La prima si chiama C, ed è una parola lunga da 5 a 14 caratteri, la seconda è una data e si chiama D, la terza è un numero, compreso tra 4.000.000 e 12.000.000 ed arrotondato a 1.000 (la terza cifra prima della virgola). Costruita

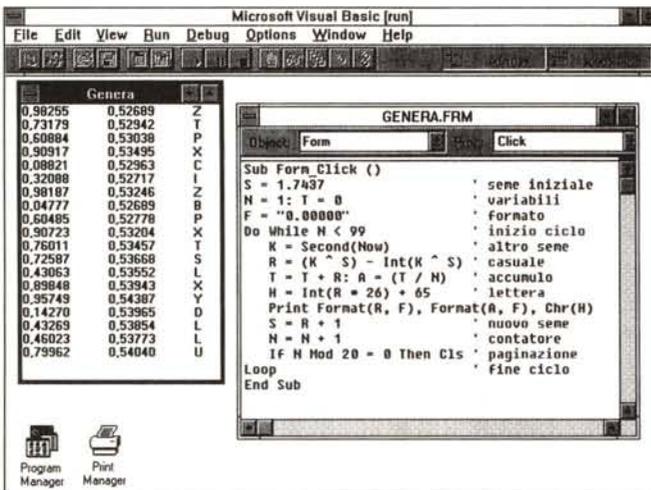


Figura 4 - VB3 - Tabella Numerica - Riempimento Casuale ed Ordinamento - Output. In Basic l'istruzione che genera il numero casuale è la RND. La usiamo per alimentare una griglia con una serie di numeri compresi tra 100 e 200. La prima colonna della griglia la riempiamo con un progressivo e analogamente la prima riga. Il riempimento viene eseguito al click sul pulsante Riempi.

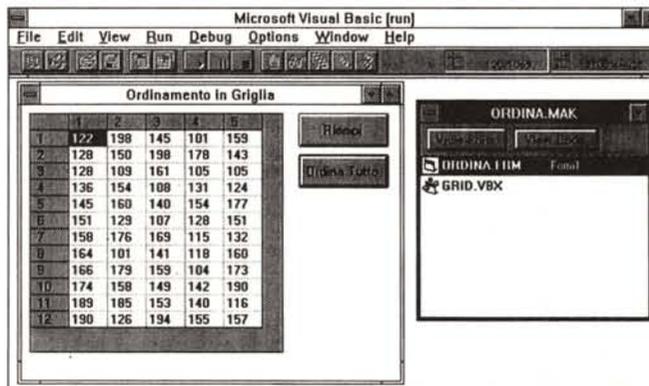


Figura 3 - VB3 - Generazione automatica di numeri casuali. Il Basic, già da un paio di decenni, dispone di una funzione (si chiama RND) che genera un numero casuale compreso tra 0 e 1 (1 escluso). C'è una istruzione aggiuntiva (la Randomize) che serve per fare in modo che la serie dei numeri casuali sia sempre differente. In certi casi applicativi serve invece che la serie sia sempre la stessa. Presentiamo una listato che mostra come generare in proprio dei numeri casuali rinunciando ai servizi di RND e Randomize.

nella vita, lo si può lasciare girare anche una notte. Calcolando 10 ore e un paio di record al secondo costruiremo, in tal modo, un file DBF con 70.000 record.

Se fosse necessario si possono, ovviamente, costruire più variabili, anche se va detto che non è necessario che tutti i campi siano differenti gli uni dagli altri, giusto i campi su cui si creano gli indici e su cui si eseguono operazioni di ricerca, gli altri possono essere anche uguali.

Il seme iniziale è S, poi, ogni volta che serve, viene eseguito il calcolo del numero casuale, con un algoritmo che usa due semi, T e S. La variabile T viene presa ogni tanto dall'orologio, e precisamente dai secondi della funzione TIME. Una volta generato il numero casuale viene ridefinito il seme.

Anche il secondo esercizio serve per generare in proprio un numero casuale. Questa volta usiamo il Visual Basic, ma solo sue istruzioni standard, presenti cioè in qualsiasi Basic.

L'algoritmo è del tutto analogo a quello visto, se lo avete visto, in dBase. Il programma viene eseguito al verificarsi dell'evento click sulla Form e visualizza tre colonne sulla Form stessa.

Nella prima il numero generato, nella seconda l'accumulo con la media dei numeri generati (che deve tendere a 0,5, anzi se non tende a 0,5 vuol dire che l'algoritmo è sbagliato), nella terza la conversione del numero in carattere alfabetico. Cosa che si fa prendendo il numero, moltiplicandolo per 26, aggiungendogli 65 e usandolo in una funzione CHR, che restituisce il carattere ASCII corrispondente.

Usiamo la funzione RND

In Basic l'istruzione che genera il numero casuale tra 0 e 1 è la RND. La usiamo per alimentare una griglia con una serie di numeri compresi tra 100 e 200. La prima colonna della griglia la riempiamo con un progressivo e analogamente la prima riga. Il riempimento viene eseguito al click sul pulsante Riempi (vedi la figura 4). E fino a qui nessuna difficoltà.

Usiamo la griglia per realizzare ed applicare una procedura di ordinamento che metta le righe della griglia nell'ordine imposto dal numero presente nella prima colonna.

Per mettere in ordine una serie di numeri o parole basta confrontarli a coppie e se non stanno nell'ordine giusto occorre scambiarli di posto. Per essere sicuri di eseguire tutti i confronti vanno impostati due cicli, uno dentro l'altro, il primo scorre tutti gli elementi e il se-

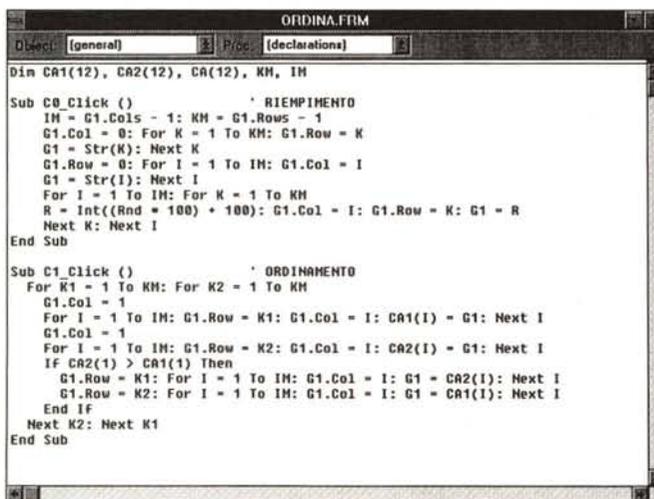


Figura 5 - VB3 - Tabella Numerica - Riempimento Casuale ed Ordinamento - Listato. Una delle routine fondamentali nella storia della programmazione è quella che serve a mettere in ordine dei dati. Nei prodotti evoluti l'ordinamento è una funzionalità sempre disponibile per cui, quando si debbono mettere a posto dei dati, basta evidenziarli e pigiare un pulsante. In VB tale funzionalità non c'è. Vediamo quindi come mettere in ordine, rispetto al contenuto della prima colonna di numeri, i dati della nostra griglia.

una serie di tre variabili la si può scaricare su un nuovo record con le due istruzioni:

```

APPEND BLANK, che aggiunge un record vuoto
REPLACE <nomecampo> WITH

```

<nomevariabile>, che riempie un campo con una variabile.

Un programma del genere, che potrebbe essere compilato con un compilatore per xBase, non è veloce, ma dovrebbe essere eseguito una sola volta...

condo anche. Questo garantisce tutti i confronti e quindi garantisce il fatto che alla fine tutti gli elementi siano a posto.

Una delle routine fondamentali nella storia della programmazione è proprio quella che serve a mettere in ordine dei dati (il famoso SORT). Quella mostrata (in figura 5) è quella più immediata e comprensibile ma è la più... suicida. Ne esistono tante altre che raggiungono lo scopo eseguendo meno confronti e quindi mettendoci meno tempo.

È chiaro che nei prodotti evoluti l'ordinamento è una funzionalità sempre disponibile per cui quando si debbono mettere a posto dei dati basta evidenziarli e pigiare un pulsante. In VB tale funzionalità non c'è, ce la siamo dovuta fare da soli.

Con il successivo esercizio, realizzato con MS Excel 4.0, verificiamo il fatto che il numero casuale sia veramente casuale. La verifica consiste nel «tirare» mille volte due dadi, contare quante volte viene il 2, quante il 3, e così via fino al 12. Eseguiti i 1.000 lanci contiamo le frequenze e le confrontiamo con il calcolo delle probabilità, che dice che il 2 esce una volta su 36, il 3 due volte su 36, e così via.

La «legge empirica del caso» dice che la probabilità (che si valuta a priori) deve tendere, aumentando il numero delle estrazioni, alla frequenza (che si calcola, a posteriore, sugli eventi). Per tirare un dado:

=INT(RND*6)+1

Per tirarne due si sommano due formule uguali.

Per calcolare la frequenza si usa la funzione corrispondente, che è di tipo «matrice» e quindi va costruita su una serie di celle. La funzione vuole due parametri, il gruppo dei dati da analizzare e la serie di valori rispetto ai quali eseguire il confronto:

=FREQUENZA(eventi;confronto)

Per chi non conoscesse le formule in matrice diciamo che vanno scritte su una selezione di celle e che alla fine va digitato Ctrl Invio, anziché un semplice Invio. La formula corrispondente, che, ripetiamo, vale su più celle, è racchiusa tra parentesi graffe.

Nella figura 6, in basso, vediamo l'inizio delle serie dei lanci dei dadi. In alto vediamo la serie rispetto alla quale abbiamo calcolato le frequenze, le frequenze assolute (numero di eventi), le percentuali, calcolate rispetto al totale degli eventi (1.000 lanci) e le probabilità. Ovviamente da tale insieme di dati traiamo un significativo grafico (figura 6). Dai numeri e dal diagramma risulta evidente l'assoluta affidabilità del generatore di numeri casuali di Excel.

Figura 6 - Excel 4.0 - Prova, con lanci di due dadi, della funzione Casuale.

In Excel 4.0, così come in qualsiasi prodotto di tipo foglio elettronico di qualsiasi generazione, è presente una funzione per la generazione di numeri casuali (nella versione italiana si chiama CASUALE!). Facciamo subito un esercizio tendente a verificare l'affidabilità di tale funzione ovvero a verificare se genera dei numeri veramente casuali. Simuliamo 1.000 lanci di due dadi. Contiamo le frequenze di uscita di ciascun numero e le confrontiamo con le corrispondenti probabilità di uscita. Le due curve non dovrebbero discostarsi troppo.

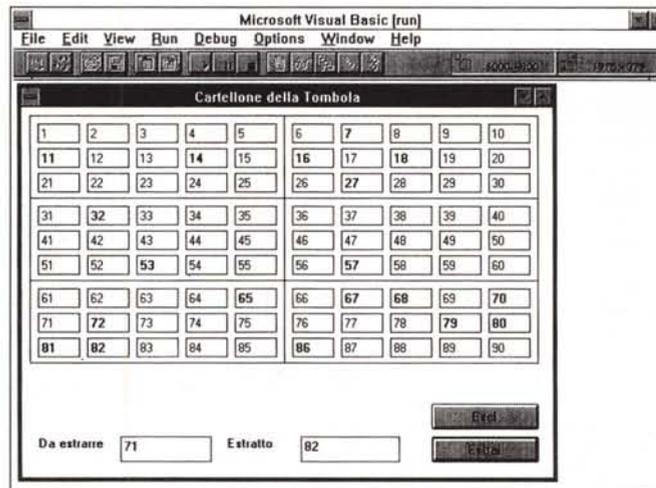
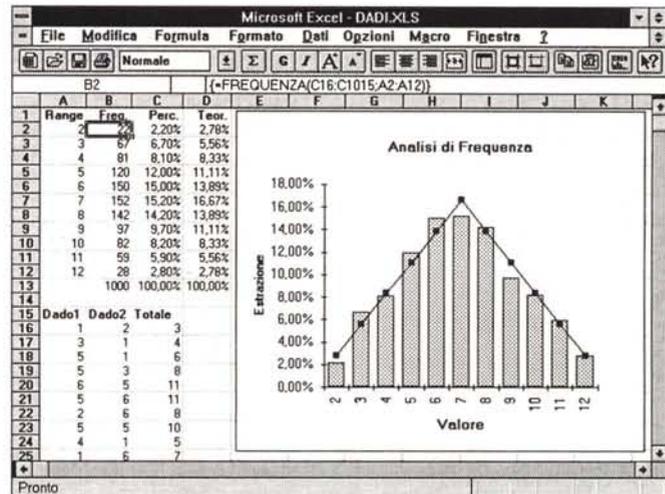


Figura 7 - VB3 - Cartellone della Tombola - Estrazione senza ripetizione - Output.

Un classico campo di applicazione del generatore di numeri casuali è quello dei Giochi, in cui gli avvenimenti dipendono più o meno da una certa casualità.

Nel lancio di due dadi può venire più volte il 12. In altri tipi di gioco, invece, vige la «regola della non ripetizione». Ad esempio in un qualsiasi gioco di carte all'inizio le carte vengono distribuite con un criterio che si basa sull'estrazione «senza ripetizione». Se una carta «esce» non può uscire una seconda volta. Nel nostro esercizio mostriamo un metodo, applicato al tabellone della tombola, grazie al quale un numero uscito non può più uscire.

Come evitare di far uscire sempre lo stesso numero

È uno dei problemi classici nella teoria dei giochi e si chiama «estrazione senza ripetizione». Nei giochi di carte, una volta uscita una carta non esce più, nella tombola una volta uscito un numero non esce più. Al contrario del gioco dei dadi in cui uno stesso numero può uscire più volte, anche consecutive.

Si tratta dunque di trovare un algoritmo elegante, che eviti che un numero (riferiamoci direttamente alla tombola) già uscito non esca più, e non che, più banalmente e scorrettamente dal punto di vista formale, controlli se un numero è già uscito.

La soluzione classica è quella di co-

struire un Array di 90 valori, da riempire inizialmente con i numeri da estrarre, ad esempio:

N(45)=45

Se esce, come primo numero tra 1 e 90, il 45 si preleva il 45.mo valore dell'Array, che rappresenta il numero estratto (il 45), e lo si sostituisce con il valore dell'ultimo elemento dell'Array:

N(45)=N(90)

e poi si decrementa il valore che indica i numeri da estrarre.

Se esce, questa volta tra 1 e 89, di nuovo il 45, si preleva il 45.mo valore dell'Array, che è il successivo numero estratto (il 90), e si procede alla solita sostituzione e al decremento.

In tal modo, abbastanza semplice, si calcola prima un numero casuale da 1 a

Figura 8 - VB3 - Cartellone della Tombola - Estrazione senza ripetizione - Listato.

I listati sottostanti i vari oggetti presenti nell'applicazione sono cortissimi. Soprattutto per il fatto che i vari numeri del Cartellone sono legati ad un Array di Control e quindi in pratica corrispondono ad un solo oggetto. Al caricamento della Form viene caricata l'Array di appoggio (il cui scopo è descritto nel testo), poi si va avanti con il click su Estrai che «pesca» il nuovo numero, sistema l'Array di appoggio e aggiorna il contatore dei numeri da estrarre. L'evento Change sull'Array delle Caselle di Testo serve per l'eventuale visualizzazione, in una Label altrimenti invisibile, del significato del numero secondo la tombola partenopea. Abbiamo previsto di «scodificare» solo il famoso 77.

```

TOMBOLA.FRM
Object: [general] Proc: [declarations]

Dim N(90), T

Sub Form_Load ()
    On Error Resume Next
    For I = 1 To 90: N(I) = I
    T(I - 1) = I: T1(I - 1).FontBold = False
    Next I
    T = 90: T4.Visible = False
End Sub

Sub E_Click ()
    Randomize:P = Int(Rnd * T) + 1: T3 = N(P)
    T1(T3 - 1) = N(P): T1(T3 - 1).FontBold = True
    N(P) = N(T): T = T - 1: If T = 0 Then End
    T2 = T
End Sub

Sub T1_Change (I As Integer)
    T4.Visible = False
    Select Case I + 1
    Case 77
        T4.Visible = True
        T4 = "le gambe delle donne"
    End Select
End Sub
    
```

90, poi un numero da 1 a 89, e così via fino ad esaurimento, mentre il numero estratto viene prelevato dall'Array.

Questo meccanismo si può utilizzare per la tombola e per tutti i giochi di carte.

Nel caso della tombola, ci riferiamo alle figure 7 e 8, il problema successivo è quello di individuare quale sia l'elemento corrispondente al numero estratto nell'Array di Text.Box, che costituisce il nostro Cartellone. La soluzione più semplice è quella di fare in modo che Array di Controls e Array di Numeri corrispondano (basta farli partire ambedue da 1).

Addensamenti casuali

Se si genera un numero casuale intero compreso tra 1 e 100, con uno dei metodi visti fino ad ora, ciascun numero ha un'eguale probabilità di uscita di ciascun altro.

In molti casi però non ci si può accontentare di una distribuzione così omogenea. Ad esempio in statistica sono riconosciute almeno una dozzina di tipologie di distribuzione. Pensate, ad esempio, alla misura dell'altezza dei bambini di 10 anni. I dati rilevati in un campione di 1.000 bambini non saranno distribuiti uniformemente dal valore più basso al più alto, ma si addenseranno attorno al valore medio, che costituirà l'altezza normale di un bambino di 10 anni.

Nel generare dei numeri casuali quin-

Figura 9 - Excel 5.0 - Numeri casuali che seguono una distribuzione Normale - Dialog Box.

In certi casi esiste la necessità di «addensare» i numeri casuali. Ad esempio potrebbero servire dei numeri che vanno da 0 a 100, ma che si addensano vicino allo 0. Un sistema potrebbe essere quello di estrarre 10 ed elevarlo al quadrato. In Excel 5.0 esistono una serie di funzionalità che permettono di «regolare il flusso» dei numeri casuali. Ad esempio numeri che seguono una Distribuzione Normale, della quale va impostato il valore della Media e il valore della Deviazione Standard.

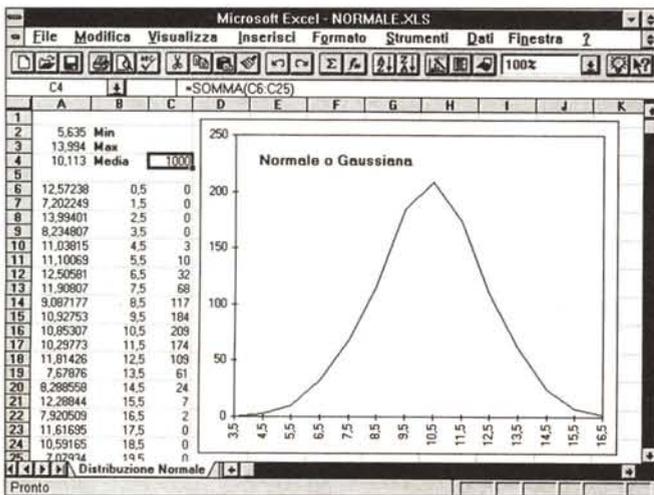
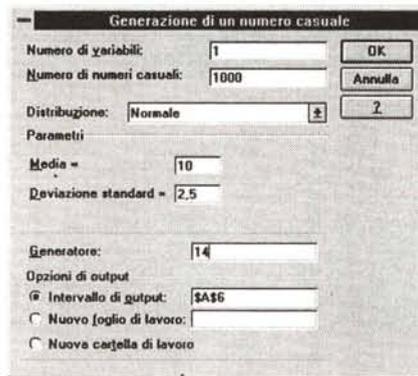


Figura 10 - Excel 5.0 - Numeri casuali che seguono una distribuzione Normale - Grafico. I numeri casuali generati si addensano attorno al 10 con una deviazione standard pari a 2.5. La Deviazione dà la misura di tale addensamento, tanto più è bassa tanto più la curva ha un picco in corrispondenza del 10. La curva risultante appare spigolosa sia perché le estrazioni non sono tante, sia perché gli intervalli di calcolo sono abbastanza larghetti.

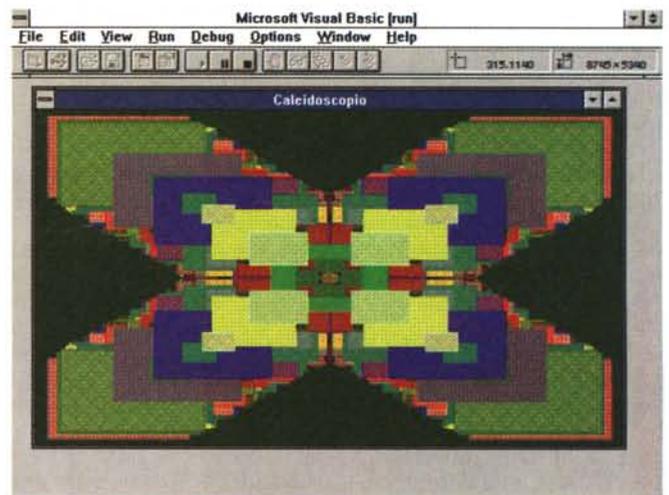


Figura 11 - VB3 - Numeri Casuali addensati per Caleidoscopio - Output. Prendete un numero casuale tra 0 e 10 e elevatelo al quadrato. È chiaro che il numero risultante può variare tra 0 e 100. Solo che tale numero si addensa vicino allo 0. Infatti se il numero è 1, elevato al quadrato rimane 1, se il numero è 5 il quadrato è 25. Allora, facendo un semplice calcolo, metà dei numeri finali saranno compresi tra 0 e 25, l'altra metà tra 25 e 100. Insomma si può regolare l'addensamento dei numeri casuali, intervenendo sulla formula generatrice.

di ci si può porre il problema di distribuirli in modo che si addensino rispetto a determinati valori.

In Excel 5 (lo vediamo nelle figure 9 e 10) è possibile creare una distribuzione di numeri casuali che segue una regola statistica. Nelle figure vediamo come creare una distribuzione normale, detta anche Gaussiana, che ha come media il valore 10, e ha come Deviazione Standard, valore che dà la misura di come la curva si addensino attorno al valore medio, pari a 2,5.

Produciamo una sola colonna (Numero di variabili) di 1.000 valori (Numero di numeri casuali). Nella successiva figura vediamo il foglio con i dati (che sono in basso a sinistra). In alto a destra riportiamo il valore minimo, massimo e medio (che è prossimo al 10), della serie prodotta. Sulla destra il classico grafico «a campana» proprio della distribuzione Gaussiana. Avendo definito pochi intervalli la curva è un po' spigolosa.

Un po' di fantasia

Gli ultimi due esercizi servono a dimostrare come, con le funzioni di generazione dei numeri casuali, sia possibile realizzare dei programmi «di fantasia», in genere programmi grafici, stile «Screen Saver» di Windows, in cui a video vengono prodotte delle linee o dei quadrati colorati, le cui posizioni, dimensioni e colori sono determinati casualmente.

Nel primo dei due esercizi (documentati in figura 11 e in figura 12) lavoriamo in un riferimento cartesiano che va da -200,-200 a +200,+200. Generiamo un rettangolo, il cui vertice di base e le cui dimensioni sono impostate in modo casuale, che viene tracciato, contemporaneamente, nei quattro quadranti. Viene così simulata una riflessione come se tra i quattro quadranti ci fossero degli specchi. Il Caleidoscopio, quello vero, in cui però gli specchi sono angolati in modo differente, funziona alla stessa maniera.

Il successivo è un programma che genera dei «Rimbalzi». Lo vediamo nelle due ultime figure, la 14 e la 15.

L'aspetto più interessante è costituito dalla routine di riconoscimento della dimensione della finestra, che serve a fare in modo che il rimbalzo sia effettuato sempre quando viene raggiunto il bordo della finestra stessa, qualsiasi sia la sua dimensione.

Lo Scale Mode (è una proprietà della Form) scelto è il terzo, quello che permette di lavorare in Pixel. Si possono provare anche gli altri modi, ma si avrà un rallentamento nel tracciamento delle

Figura 12 - VB3 - Numeri Casuali addensati per Caleidoscopio - Listato.

Con i numeri addensati ci si fanno i caleidoscopi, effetti visuali basati su concetti di casualità e di riflessione. Il riflesso lo simuliamo lavorando in uno spazio cartesiano facilmente divisibile in quattro e riproducendo lo stesso elemento rettangolare nei quattro quadranti.

```

C:\WBCORSO\EVB30.FRM
Object: Form      Proc: Load

Sub Form_Load ()
  Form1.ScaleMode = 0
  Form1.Scale (-200, -200)-(200, 200)
End Sub

Sub Form_Click ()
  For I = 1 To 100: Randomize
    J = (Rnd * 10) ^ 2: H = J + (Rnd * 10) ^ 2
    C = Rnd * 10000000
    Line (J, J)-(-H, H), C, BF
    Line (-J, J)-(-H, H), C, BF
    Line (-J, -J)-(-H, -H), C, BF
    Line (J, -J)-(-H, -H), C, BF
  Next I
End Sub

```

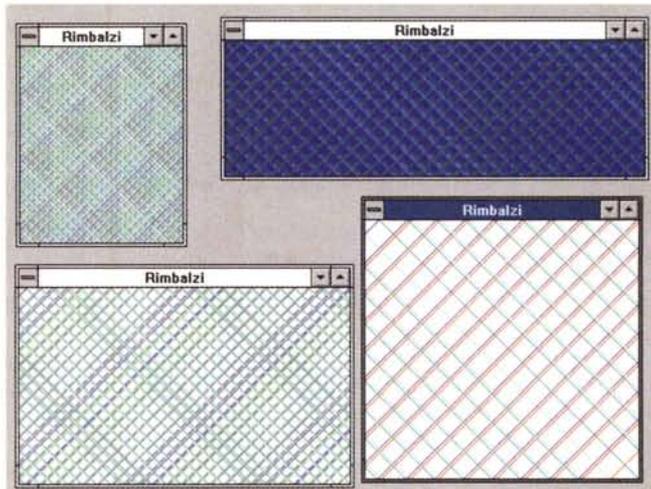


Figura 13 - VB3 - Rimbalzi Casuali - Output. Quasi uno Screen Saver. Realizziamo un programma che, in modo casuale, genera delle linee colorate che rimbalzano sulla finestra attiva. In questo caso la casualità può essere usata per definire il punto di partenza della linee, gli angoli di rimbalzo, i colori delle linee, ecc. Ci siamo limitati al punto di partenza.

Figura 14 - VB3 - Rimbalzi Casuali - Listato. Al di là dell'uso della funzione RND l'esercizio serve per capire le modalità di gestione della grafica nelle Form Visual Basic. Si può lavorare in Pixel (è la cosa migliore per applicazioni che sfruttano la grafica a video), si può «leggere» la dimensione della finestra attiva, si può tracciare una linea punto per punto. Ad intercettare le collisioni con i bordi ci dobbiamo pensare noi.

```

Microsoft Visual Basic [design]
File Edit View Run Debug Options Window Help

Rimbalzi
RIMBALZI.FRM
Object: Form      Proc: Click

Sub Form_Click ()
  Cls: i = 0: j = 0
  scalemode = 3
  l = scalewidth: h = scaleheight
  Randomize: i = Rnd * l: j = Rnd * h
  x = 1: y = 1
st: PSet (i, j), RGB(0, 81 * (x + 2), 81 * (y + 2))
  If i > l Then x = -1: DoEvents
  If j > h Then y = -1: DoEvents
  If i < 0 Then x = 1: DoEvents
  If j < 0 Then y = 1: DoEvents
  i = i * x: j = j * y
GoTo st:
End Sub

```

linee, se si lavora con una unità di misura più piccola (ad esempio i Twips), una velocizzazione e delle linee più rade, se si lavora con unità superiori (ad esempio i centimetri).

L'ultima cosa da dire riguarda la presenza della istruzione DoEvents, che

serve per lasciare Windows libero di eseguire altri processi. Questo consente, all'interno di una singola finestra, di seguire il tracciamento delle linee passo passo, e, soprattutto, di permettere a più finestre di lavorare in contemporanea, come appare nella figura.