Matematica - Scomposizione in fattori primi

Author : flavio

Categories : Excel, Matematica

Date : 20 febbraio 2013

Flavio Bertamini ci ha mandato questo file che permette di lavorare con i numeri primi:

Permette infatti di:

- Elencare i primi N numeri primi
- Scomporre un numero qualisasi in fattori primi

La seconda ovviamente è una procedura ben più complicata della prima.

Il file è ben fatto e quindi rinnoviamo i complimenti per Flavio!

Di seguito la gentile mail che accompagna il documento:

nel ringraziarLa per la mia prima pubblicazione sul Vostro sito, Le mando la versione definitiva del mio lavoro sui Numeri primi, che alla fine direi è abbastanza "gustosa".

Ho aggiunto un paio di fogli con varianti che mostrano come si possa dare ad un'applicazione Excel una veste più elegante tramite l'uso dei controlli Vba ed infine c'e' un foglio chiamato Esercizio in cui si mostrano alcuni risultati ottenuti in poco tempo mentre facevo un piccolo testing e scrivevo alcune note per chi intende programmare in Vba.

C'e' anche un foglio di presentazione. Le note si trovano nel file Articolo.doc (che non contiene macro) che potrebbe rappresentare il giusto commento al lavoro svolto.

Dal punto di vista dell'implementazione del codice ho aggiunto un Modulo2 che contiene una Function la quale serve a riordinare meglio le cose e permette di scrivere rapidamente il codice che gestisce appunto il foglio Esercizio.

Se volete fate girare un po' il programma prima della definitiva pubblicazione per verificare che non mi sia sfuggito qualche bug ne sarei lieto. Magari potete ottenere dei risultati "interessanti".

In alcune note nei fogli di .xls ci sono dei risultati che ho ottenuto e che possono essere cancellate per l'uso del foglio di calcolo, così come il grafico ma possono rappresentare una presentazione per chi intende dare una valutazione "rapida" del valore del foglio stesso.

Presentazione

Attraverso questo testo si intende portare l'utente a comprendere meglio l'uso di Visual Basic for Application: uno strumento in grado di potenziare notevolmente ma in modo semplice le nostre capacità di operare con il personal computer. Dopo una necessaria premessa teorica, il più possibile limitata, si procede all'illustrazione di alcuni esempi pratici nell'intento di favorire la presa di contatto del lettore con la materia della programmazione.

Lo scritto intende fornire all'utente, che già conosce abbastanza bene almeno l'uso Excel® standard e di Office® in generale, un potente strumento utilizzabile sia in casa che al lavoro che rende in qualche modo più "padroni" del personal computer e rappresenta un valido modo per avvicinarsi al mondo della programmazione.

Non è adatto a chi considera Excel® un semplice gestore di tabelle.

Ai Lettori:

Questo scritto intende offrire a chi è interessato un esempio sulle tecniche di programmazione più semplici che si trovano attualmente a disposizione in Excel®. L'utente ideale è una persona che conosce abbastanza bene l'uso dei principali applicativi, che ha già cominciato a seguire qualche "Tutorial" su almeno uno dei principali linguaggi oggi diffusi, ma è rimasto deluso perché si è reso conto che il mercato offre "facili" promesse di insegnamento attraverso la rete, i libri, i corsi didattici, ma alla fine le capacità informatiche non sono aumentate, se si esclude la scrittura del programma "Hello Word" o la colorazione di qualche Form in un orrido verde pisello!

In realtà la programmazione è un' arte, e, come ci invitava lo psicanalista Erich Fromm nella sua celeberrima opera "L'arte di amare", per diventare maestri nella propria arte non ci sono "vie regie" ma bisogna amare la propria arte e, per riuscire nell'intento di scrivere un programma interessante, bisogna pensare che la via può essere lunga, molto lunga. Premetto che lo scopo finale di questo scritto non è quello di insegnare un linguaggio di programmazione, ma di dire alcune cose sulla "programmazione" in generale, viste con gli occhi di un semplice autodidatta!

Le persone realmente interessate capiranno facilmente che c'è una differenza sostanziale. Un linguaggio di programmazione può essere imparato in pochi mesi ma per "programmare" non basta una vita. Tra l'altro, la programmazione è un'attività essenzialmente comunitaria, che crea enormi interdipendenze, non si può pensare di essere "veramente" dei professionisti lavorando isolati, ma certamente, mentre si stà imparando o si scrive del "codice", stare da soli è necessario per avere la giusta concentrazione.
In questa solitudine avere un "amico" che ci dà dei suggerimenti può essere utile, e lo scopo del testo è proprio questo…

I mezzi informatici necessari per la comprensione di questo testo non sono straordinari. Basta in realtà avere sottomano il foglio di calcolo Excel® per avere uno strumento molto potente, basta saperlo usare. L'infomatica si sviluppò utilizzando "schede perforate" di cartoncino come memorie per dati e programmi, e per chi, come me, ha incominciato con il mitico Fortran77 a

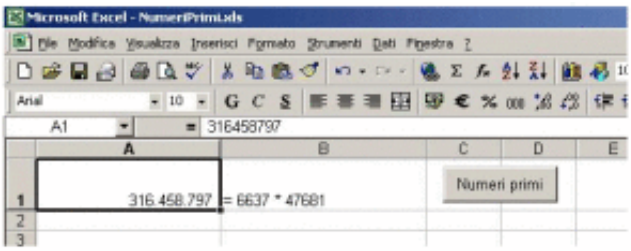
riga di comando, lo sviluppo che ha potuto vedere nel corso degli anni è stato enorme. Anche il semplice “Blocco Note” in dotazione con Windows® sarebbe stato, allora, un lusso inimmaginabile. Nessuno intende chiedervi “ora” di procurarvi una suite professionale mentre state imparando cosa significa programmare, per chi se la sente ci sono a vostra disposizione nella rete molti strumenti, ma l’umiltà, l’impegno, la costanza e la tenacia non si trovano “facilmente” nel Web. Ma gli ignoti “maestri” che hanno dipinto le cappelle preistoriche dell’Europa continentale non avevano certo “pennelli” o luci straodinarie, ma ci hanno comunque lasciato la testimonianza stupefacente delle loro capacità artistiche.

Insomma “programmare” non è una questione di “pennello”…
Non pretendete di avere da questo scritto delle dritte su quale settore della programmaziione iniziarvi o su quali linguaggi siano migliori di altri, sono domande che rivelerebbero l’ingenuità di chi le porge: sarà la vostra intelligenza semmai a suggerirvi come operare.

Non pretendete neppure qualche dritta che vi risolva un problema contingente: per questo c’è innanzitutto il “mitico” tasto F1 e poi la rete Internet. Infine non contate molto sull’appoggio degli altri, di solito chi tenta di evolvere viene irriso e chi produce dei risultati viene visto con una certa invidia.

Un primo esempio: Calcolo con i numeri primi

Aprendo il file NumeriPrimi.xls ci troviamo di fronte ad prima vera applicazione. I vari fogli ci mostrano un utilizzo molto semplice di Vba per Excel®, che richiede conoscenze matematiche molto elementari: il calcolo dei numeri primi. Diciamo molto brevemente, per chi se lo è scordato, che un numero primo è un numero che può essere sottoposto a divisione intera solo per se stesso o per uno (interessante l’articolo su Wikipedia correlato all’argomento). In genere è un problema che si affronta nella scuola media inferiore.



Immaginiamo però di voler sapere “rapidamente” come si scompone in fattori primi un numero come quello in figura nella cella A1 di Excel nell’illustrazione sopra: la cosa potrebbe richiedere un tempo irragionevole a mano o la scrittura di centinaia di celle piene di formule con il semplice Excel.

Con Vba per Excel le cose cambiano notevolmente.

In sintesi il nostro programma iniziale dovrà leggere un valore numerico nella cella A1 e ne fornirà la scomposizione in fattori primi nella cella B1.
Come si realizza un simile procedimento? Affrontiamo il problema per gradi: che cosa significa innanzitutto, in termini informatici, che un numero è primo?
Possiamo proporre la realizzazione di una funzione (una Function) che risponda a questa domanda ripercorrendo il classico procedimento imparato alla scuola media.
Infatti il primo passo nel nostro sviluppo è quello di scrivere questa funzione: apriamo dunque l'Ide (Integrated Development Environment) di Vba (con il comando Alt + F11) ed analizziamo il Modulo1 (per aprire cliccare due volte sull'icona della pagina correlata nella finestra "Progetto").

Il Modulo1 contiene delle routines che vengono utilizzate in tutto il foglio di calcolo.
Incominciamo a leggere:

Option Explicit: è un istruzione che serve ad indicare al compilatore che ci si servirà della dichiarazione esplicita delle variabili, è sempre bene usare questa formulazione quando si programma anche se non è strettamente obbligatorio, in tal modo si evitano spesso degli errori grossolani che generano fastidiosi problemi in fase di debug.

Option Base 1: questa istruzione indica al compilatore che i nostri array partono con l'indice 1 e non zero come altrimenti accade.

Public Primi(5000) As Double : Primi() è un array che contiene cifre a doppia precisione e ci serve per immagazzinare i primi 5000 numeri primi.

In seguito analizziamo la function isPrime che rappresenterà il cuore del nostro programma .

```
Public Function isPrime(ByVal variabile As Double) As Boolean   '-----
-----------------------------------   ' determina se un numero e' pri
mo   '----------------------------------------       Dim isPri As Boo
lean       Dim Modo As Integer       Dim i As Integer         isPri =
True       For i = 2 To 5000           Modo = variabile Mod Primi(i)
         If Modo = 0 Then               isPri = False               Ex
it For           ElseIf Primi(i) * Primi(i) > variabile Then
     Exit For           End If       Next       isPrime = isPri    End
 Function
```

Cosa dice la nostra routine?
Public significa che la routine è visibile in tutto il nostro progetto; isPrime legge la "variabile" come doppio, ed effettua un confronto con l'array Primi() nel ciclo For.
Il valore restituito è un booleano (Vero/Falso).
Vediamo di seguito come nella memoria del Pc viene rappresentato l'array Primi().

Elemento	Nome	Valore
1	Primi(1)	2
2	Primi(2)	3
3	Primi(3)	5
4	Primi(4)	7
.....	....	.....
5000	Primi(5000)	48.611

L'array Primi() conterrà quindi le cifre 2,3,5….fino a 48.611 cioè i primi 5000 numeri primi: supponiamo per il momento infatti che l'array sia già inizializzato. Come procede la nostra routine? All'interno del ciclo For valuta la variabile "Modo" che rappresenta il resto della divisione intera tra "variabile" e l'elemento dell'array Primi(i), attraverso l'operatore Mod.
La variabile booleana isPri è impostata su Vero in ingresso alla routine.
Se "Modo" =0 significa che il mumero non è primo e quindi la nostra variabile isPri sarà falsa.
Si può uscire quindi dal ciclo con l'istruzione: Exit For.
Un altro modo per uscire dal ciclo For si realizza quando il quadrato di Primi(i) è maggiore della variabile in oggetto. In tal caso non e' più possibile avere divisioni intere per la nostra variabile che sarà certamente un numero primo (isPri e' già impostato in tal senso).
Uscita dal ciclo For il valore della nostra function viene fornito dalla variabile isPri.
Notate che il ciclo inizia con i=2 e questo significa che i numeri pari non vengono valutati (Primi(1)=2; si suppone infatti che si sappia che i numeri pari non sono primi e si controlli il caso altrove).
Vediamo così che il "nocciolo" matematico si risolve in appena 14 righe di programma.
Rimane da inizializzare l'array che contiene i nostri 5000 numeri primi.
Questo viene realizzato con una Sub chiamata naturalmente Inizializza() che sfrutta a sua volta la Function isPrime.
In sintesi la Sub inizializza "puntualmente" i primi due elementi dell'array Primi() con le istruzioni:

```
Primi(1) = 2    Primi(2) = 3
```

Utilizza quindi una variabile temporanea chiamata Start che assume tutti i valori dispari cominciando da 3 e che viene incrementata di 2 ogni volta che viene eseguita l'istruzione: Start= Start +2.
Notate che il ciclo succesivo For incomincia con i=3 fino a 5000, ma le righe di codice interne vengono eseguite molte più volte perché all'interno del ciclo agisce pure l'istruzione GoTo inizio, che rimanda l'esecuzione all'incremento di Start, e viene letta tutte le volte che il numero analizzato NON è primo senza con questo incrementare la variabile i che controlla il ciclo stesso.
Le poche righe di codice cosa fanno? Trovano i primi 5000 numeri primi.
Le variabile Start viene analizzata dalla function isPrime e se risulta un numero primo viene immagazzinata nell'array Primi() altrimenti viene incrementata di 2.

Il primo valore analizzato, notate bene, è 5 che risulta essere un numero primo.
La variabile booleana pubblica Done viene impostata sul valore “True” alla fine della routine, per indicare al nostro programma che il calcolo dei primi 5000 numeri primi è stato effettuato ed è residente in memoria.
Notate che non c’è contraddizione per il fatto che le due routines così realizzate sembrino mangiarsi la coda l’una con l’altra: infatti la ruotine Inizializza() prima di richiamare la funzione isPrime ha già immagazzinato il numero minimo di elementi (il solo numero 3) che permettono di operare al nostro software.
Non si hanno problemi perché il ciclo For non viene mai completato in questa fase.
Realizzate queste due piccole routines che serviranno in tutto il nostro foglio di calcolo possiamo passare all’utilizzo delle stesse in alcune applicazioni che mostriamo di seguito.

Foglio: Tabella.

Nel foglio chiamato “Tabella”, vogliamo inizialmente far scrivere ad Excel alcuni numeri primi sul foglio di calcolo per controllare rapidamente la validità delle nostre implementazioni.
Mettiamoci in modalità “Progettazione” cliccando sull’apposito pulsante. Creiamo quindi un pulsante impostando la proprietà Name al valore BtnTabella e la Caption al valore Tabella. In fase di progettazione possiamo cliccare due volte sul pulsante realizzato, per creare nello Ide di Visual Basic for Application la Sub BtnTabella_Click() che si scatenerà in fase di esecuzione al cliccare sul pulsante stesso.
Notiamo che la Sub così creata è dichiarata Private cioè non è visibile in tutta l’applicazione.
La Sub è molto semplice: dopo aver verificato che la Sub Inizializza() sia stata eseguita, ripulisce il foglio di calcolo dai dati eventualmente scritti precedentemente, e quindi propone un InputBox che ci chiederà il numero di numeri primi che vogliamo visualizzare sul nostro foglio Excel. All’ ok dato all’Input Box seguirà un ciclo For di scrittura nelle varie celle della prima colonna del foglio “Tabella” con completamento finale di un breve titolo. Si consiglia di non salvare troppi dati in fase di chiusura del file .xls anche perché si possono recuperare con un semplice clic. Può risultare comodo il pulsante “Cancella” che con una routine di una sola riga ripulisce il foglio. Indubbiamente sono delle routines che non producono risultati sorprendenti, ma è un utile esercizio iniziale.

Foglio: Scomponi in fattori primi

Nel foglio “Scomponi …” si userranno le due rotines del Modulo1 per scomporre un numero in fattori primi.
La routine CmdScomponi_Click() che viene scatenata tutte le volte che si preme il CommandButton nel foglio durante l’esecuzione.
Per visualizzarla basta premere due volte col tasto sinistro del mouse nella finestra “Progetto” nell’ Ide di Vba sull’icona del foglio di calcolo.
Cosa fa questa routine? Scompone un numero in fattori primi.
Per prima cosa si accerta che la Sub Inizializza() sia stata eseguita attraverso la lettura della variabile booleana Done (che appunto per questo è stata dichiarata pubblica) e nel caso procede alla sua esecuzione. Sappiamo quindi a questo punto, che nella memoria del nostro computer ci sono i primi 5000 numeri primi.
La routine procede all’analisi della variabile Str in cui è stato provvisoriamente immesso il

valore della cella A1 e si accerta che non sia un numero con virgola perché in tal caso le operazioni indicate in seguito non avrebbero senso. Tale controllo viene eseguito dalla funzione Instr. In caso di numero con virgola manda un messaggio di avviso all'utente tramite l'istruzione MsgBox e procede alla terminazione della routine. Se invece è stato correttamente immesso un numero naturale questo viene immagazzinato nella variabile Start come doppio.
In seguito procede alla dichiarazione dell'intercettazione degli errori attraverso la dichiarazione: On Error GoTo Errtrap che sposta la gestione di eventuali errori in fondo alla routine.
Il calcolo vero e proprio è costituito dal ciclo For che confronta la variabile Start con i vari numeri primi che abbiamo inizializzato. In Rest è posto il risultato dell'operazione Start Mod Primi(i). Se Rest =0 allora Primi(i) è un divisore di Start: la variabile j, inizializzata a zero, verrà incrementata di 1, l'array Sco() verrà ingrandito di un elemento salvando gli elementi già presenti con l'istruzione ReDim Preserve Sco(j) e conterrà all'interno di Sco(j) un valore pari all'elemento Primi(i), quindi Start verrà diviso per questo numero, la variabile ottenuta sarà chiamata ancora Start e sarà necessario ripartire dalla linea chiamata "inizio" con il nuovo Start ottenuto, per procedere ulteriormente nella fattorizzazione. Nel caso poi che il quadrato di Primi (i) superi il valore di Start, allora Start è un numero primo.
La scomposizione del nostro numero è a questo punto completata nell'array Sco(): con però un piccolo difetto: infatti l'ultima divisione potrebbe avere come quoziente 1. A tale inconveniente provvede l'istruzione:

```
If Sco(j) = 1 Then                ReDim Preserve Sco(j - 1)  End If.
```

Tale istruzione semplicemente elimina l'ultimo fattore di Sco() se questo fattore ha come valore il banale uno.
Non resta che mettere in forma leggibile i vari fattori scrivendo la formula come stringa nella variabile Str che verrà immessa in seguito nella cella B1.
L'istruzione Exit Sub costringe quindi Excel ad uscire dalla procedura mentre le istruzioni succesive vengono eseguite solo in caso di errore e consistono nella segnalazione tramite messaggio critico opportuno e succesivo "Clear" dell'errore stesso.
In tal modo l'utente occasionale non viene coinvolto da Excel nel richiamo standard del debug dell'applicazione.
Se il numero in questione è primo allora il calcolo porterà ad un array Sco() che contiene un solo elemento: in tal caso

```
Ubound(Sco())=1
```

e nella cella B1 verrà visualizzata la scritta "Numero Primo".

In caso contrario la variabile stringa Str viene inizializzata nella forma ('=) e ad essa vengono poi aggiunti i vari fattori separati da asterischi(*).
Una riprova della validità del nostro programma stà nel fatto che eliminando manualmente l'apice all'inizio della stringa che compare come output nella cella B1 (operando questo tramite

Excel), si ottiene una formula di scomposizione vera e propria che viene letta da Excel e darà naturalmente come risultato il numero di partenza.
Facciamo notare inoltre che sebbene i numeri primi immagazzinati siano “solo” 5000 il nostro programma analizza cifre ben più grandi scomponendo in fattori ad esempio numeri come 21.659.636 = 47 * 460.829 dove il secondo fattore è circa 10 volte più grande di Primi(5000). In generale possiamo dire che 5000 numeri primi sono in grado di soddisfare la curiosità dei dilettanti della matematica, ma nulla vieta, in linea di principio di impegnare Excel e il Visual Basic su terreni più “tortuosi” aumentando la dimensione dell’array Primi()! La dimensione massima possibile di un array varia tuttavia a seconda del sistema operativo in uso e della memoria disponibile. L'utilizzo di array di dimensioni superiori alla memoria RAM disponibile nel sistema comporterà un rallentamento notevole del funzionamento poiché i dati devono, in tal caso, essere letti e quindi scritti su disco. La dimensione, inoltre, è stata scelta anche per rendere il programma abbastanza veloce.

Foglio:Uso dei controlli

Nel foglio “Uso dei controlli” si dà al nostro piccolo programma di calcolo per la scomposizione dei fattori primi una versione grafica più accattivante con l’uso dei controlli Label e TextBox. La scrittura del codice non richiede molto tempo perché è sostanzialmente la stessa del foglio precedente solo che mentre nel primo avevamo come oggetti le celle di Excel ora abbiamo questi due nuovi controlli. Si può quindi ottenerla facilmente copiando la routine precedentemente sviluppata nel foglio ”Scomponi..” e successivamente apportando le opportune correzioni alla routine in meno di 10 minuti. Il foglio però assume un aspetto decisamente più “professionale”…

Foglio: Esercizio1 (e Modulo2)

Nell'implementazione del codice nei due fogli precedenti si è creata la potenzialità di avere una potente function che scompone in fattori un numero…
Questa function è stata implementata rapidamente e sviluppata a partire dal codice del foglio precedente “Uso dei controlli”, e “sistemata” nel Modulo2 per comodità ed è stata chiamata ScomponiinFattoriPrimi. Invece un Variant ed emette una stringa attraverso le varie uscite…

Questa function è però sufficientemente potente da permettere di scrivere una ruotine che calcola un “Esercizio” di scomposizione in fattori primi in appena 5 righe di codice…….

Considerazioni conclusive

Questo breve esempio ha solo un valore “didattico” e può essere interpretato in molti modi… Questo “software” può essere realmente utile a run-time solo ad un bambino che non ha voglia di svolgere gli esercizi di matematica…..

Chi si occupa da un po’ di Excel® “professionalmente” riconoscerà anche una formattazione ed uno stile dei fogli di calcolo che però è volutamente “dimesso”. Alcune considerazioni diverse potranno esserci per per coloro che intendono apprendere le basi della programmazione…

Una lettura iniziale non comporta la messa in esecuzione delle macro.
Innanzitutto guarderanno come si presentano i vari fogli con una certa calma, e quindi si porteranno con AltF11 a leggere il codice “prima” di mandarlo in esecuzione e cercheranno di capire bene…se possa avere un valore.

Certamente il codice può essere migliorato in molti modi…

Forse ad alcuni sorgerà l’idea di potenziare la Sub di inizializzazione Inizializza nel Modulo1 portando il valore da 5000 a valori più elevati……per vedere come migliorano le prestazioni di Excel nel calcolo…e cercare un prestazione migliorata.

Interessante sarebbe anche inserire una possibilità di lasciar decidere all’utente in apertura del Foglio di calcolo come inizializzare i nostri procedimenti… Questo può essere facilmente implementato scrivendo la seguente:

```
Private Sub Workbook_Open()          Inizial = InputBox("A quale valor
e desideri inizializzare il calcolo?", "Numeri Primi")   ‘   ‘   End S
ub
```

Naturalmente bisogna dichiarare la variabile Inizial. L’autore la lascia con gli apici di commento……..Bisogna inoltre sostituire Inizial dove c’e’ scritto 5000 nelle varie routines…

Ad altri forse potrà interessare l’idea di migliorare la Function che ad esempio dà l’inelegante 128=2*2*2*2*2*2*2 con una scrittura più sintetica del tipo 128= 2^7 e che coinvolge le potenze nell’espressione della nostra stringa.

Certamente modificare e potenziare il codice di un altro non è cosa semplice senza un’adeguata comprensione del problema che intende risolvere….

Per facilitare il compito della comprensione del codice sono stati messi dei commenti tra le righe di codice.
A questo proposito c’è un breve commento nel codice che recita:

```
 '-----------------------------------------------------------------
  'per i più smaliziati: provate a scrivere a questo punto    'la riga
 seguente così: Str = "=" e mettete in esecuzione   'basta cancellare
un apice eppure...   '-----------------------------------------------
------------------                              Str = "'="
```

Questo commento si trova nel Modulo 2.

E’ una specie di indovinello per divertire gli appassionati: Excel® è sempre molto

attento…anche agli apici.

La difficoltà principale che si incontra nello scrivere questo programma sta, paradossalmente, nella gestione dei piccoli numeri come 1 che solo in tempi relativamente recenti e' stato considerato"non primo".

Infatti la gestione dei "grandi numeri" è un compito che esula dal noto foglio di calcolo.
Gli "smanettoni" potranno divertirsi a far calcolare al programma numeri sempre più grandi.
Faccio notare, infine, che il codice scritto nei due Moduli può facilmente essere esportato agendo sul Menu "File" alla voce " Esporta File" ed in tal caso si hanno due moduli con estensione bas pronti sul vostro computer che possono essere caricati ad esempio su Vb6 ma anche, con qualche ritocco, su Visual Basic 2008.

Questo però esula dalle considerazioni di questo scritto che è rivolto solo ai conoscitori di Excel.

File di esempio:

NumeriPrimi2