di Raffaello De Masi

Ancora sui numeri primi

Sarebbe un peccato lasciare l'argomento numeri primi ingabbiato tra le sbarre del pur valido Crivello di Eratostene, del quale abbiamo parlato la volta scorsa. Questo è concettualmente abbastanza semplice, facile da utilizzare ed è molto utile per chi desidera una lista di numeri primi compresi tra il 2 ed un qualsiasi altro numero (o, adeguando il programma pubblicato nella precedente puntata, quelli intercorrenti tra due numeri qualsivoglia). Ma il problema può presentare una diversa piega, probabilmente anche più realistica. La domanda potrebbe essere, infatti: «Il numero XXXXXXXX è o no primo?».

La via più semplice, banale (e lunga) per determinare se un numero è davvero primo è quella di verificare se diano resto tutte le divisioni possibili tra il numero stesso, che chiameremo N, e tutti i numeri più piccoli di N, esclusa ovviamente l'unità ed il numero stesso. Il programma della figura 1 esegue proprio questo.

Esso è concettualmente semplice e piuttosto banale: viene testata la divisione tra N e tutti i numeri ad esso inferiori controllando se N (5 gruppo di istruzioni del programma, contrassegnate dalla label «3:») possa essere pari al prodotto di un numero arbitrario (introdotto dal loop) per il valore intero del rapporto di N e dello stesso numero; routine semplice e di facile implementazione.

Il programma è efficace, ma per niente efficiente. A parte la rozzezza e la brutalità della routine di soluzione, i tempi di esecuzione si allungano enormemente con cifre molto grandi. Tanto per fare un esempio, il programma allegato, redatto in Microsoft Basic ed utilizzato su un Macintosh, ha impiegato per testare la primarità di 9001 ben un minuto e mezzo (un tempo pressoché eguale su un PC IBM e una manciata di secondi in meno su un Hewlett-Packard 87XM). Ma siamo solo agli inizi della ricerca dell'algoritmo ottimale.

In effetti tutti mi avranno già dato dell'untore per il fatto che è inutile testare se N è divisibile per un numero superiore ad INT (N/2) (un numero non può essere divisibile per un altro che sia più grande della sua metà). Ma possiamo ancora affinare la nostra tecnica. Infatti è inutile testare la divisibilità del numero N per numeri superiori al valore (intero, approssimato) della radice quadrata di N. È infatti chiaro che, se M è un intero che divide N ed è più grande di INT (SQR(N)) allora N/M è un intero che divide N ed è più piccolo di INT (SQR(N)); in altre parole, a livello forse più intuitivo, la radice quadrata di N rappresenta il valore medio oltre e qualunque coppia di fattori, con prodotto N, è rappresentato almeno da un numero inferiore al valore intero della radice quadrata.

Ma non basta: la nuova versione del programma può essere ulteriormente migliorata utilizzando una tecnica mutuata dal Crivello di Eratostene; eliminiamo cioè dal test tutti i numeri pari ed ogni terzo numero dopo il 3. Tutte queste migliorie hanno portato ad un notevole incremento della velocità intrinseca del programma. La verifica dello stesso numero ha stavolta richiesto meno di un secondo su Mac ed è stata pressoché istantanea su HP (0,6 secondi). Un numero più grande, come 3519239 ha richiesto 9 secondi (sempre sul Mac); i tempi, ovviamente, divengono lunghi con numeri piuttosto grandi; sempre nelle stesse ipotesi precedenti il test di 833316667, primo anche lui, ha richiesto la bellezza di 122 secondi. La versione 1.1 della figura 2 mostra le migliorie apportate.

Un ulteriore, definitivo affinamento al programma è dato dall'aggiunta di una subroutine che consente di evidenziare, nel caso di numeri non primi, i fattori primi del numero stesso. Il listato 3 incorpora tale ulteriore capacità.

I listati, proprio con l'intento di essere chiari, non indulgono ad alcuna finezza ed eleganza operativa o di output.

Il secolo XVII fu uno dei più fecondi per quanto attiene alle scienze esatte. Infatti il suo nascere fu subito costellato dalla comparsa di nomi di prim'ordine nel campo della matematica, della geometria, delle scienze naturali (Nepero, Bürgi, Cavalieri, Torricelli, Cartesio, Pascal, sono solo nomi a ca-

```
        Programma di test dei numeri primi

                versione 1.0

        Il programma esegue un test sul numero in input
        verificando il risultato di una serie di divisioni successive
        tra il numero oggetto di test ed una successione
        comprese tra 2 ed N-1

1
    CLS
    WIDTH 60
    PRINT " Test per la verifica dei numeri primi"
    PRINT
    PRINT "          ------------ "
    PRINT

2
    INPUT " Indicare il numero che si intende testare  "; numero
    IF numero<3 OR numero <> INT(numero) THEN PRINT "attento,un numero valido!" GOTO 2

3
    num = numero-1 : non$ = ""

    FOR loop = 2 TO num
      IF numero = loop * INT (numero/loop) THEN non$ = "non " : loop = num
    NEXT loop

4
    PRINT " Il numero "; numero ; non$ ; " è primo."

5
    PRINT
    INPUT " vuoi ricominciare (S/N) "; risposta$
    IF UCASE$(risposta$) = "S" THEN 2

6
    END
```

Figura 1 - Primo tipo, inefficiente, di test per la verifica di un numero primo.

```
'        Programma di test dei numeri primi
'
'                  versione 1.1
'
'        Il programma esegue un test sul numero in input
'        verificando il risultato di una serie di divisioni successive
'        tra il numero oggetto di test ed una successione
'        compresa tra 2 ed n-1
'
1
    CLS
    WIDTH 60
    PRINT " Test per la verifica dei numeri primi"
    PRINT
    PRINT "              ------------ "
    PRINT
'
2
    INPUT " Indicare il numero che si intende testare  ", numero
    IF numero < 3 OR numero <> INT(numero) THEN PRINT "attento, sui numeri validi! " GOTO 2
'
'
3
    non$ = "" : num = SQR(numero)
    IF numero=2*INT(numero/2) OR numero = 3*INT(numero/3) THEN non$="non" ELSE GOSUB 10
'
4
    PRINT " il numero ", numero , non$ , " è primo "
'
5
    PRINT
    INPUT " vuoi ricominciare (S/N) ", risposta$
    IF UCASE$(risposta$) = "S" THEN 2
'
6
    END
'
'
    ************
'
'
10
    FOR loop = 5 TO num STEP 6
      IF numero = loop*INT( numero/loop) THEN non$ = " non " : loop = num
      IF numero = (loop+2)*INT( numero/(loop +2)) THEN non$ = " non " : loop = num
    NEXT loop
'
    RETURN
```

Figura 2 - Secondo tipo, ben più efficiente, di test per la verifica di un numero primo.

so in «un cielo schizzato d'argento», come ebbe ad esprimersi Eulero nelle sue Istitutiones Calculi Diferentialis). Ma accanto a tali personalità d'eccezione il secolo ospitò personaggi più modesti, piccoli ed oscuri pensatori cui non è stata mai attribuita alcuna grande scoperta. Ma la loro presenza è altrettanto importante per un ruolo insostituibile che ebbero nella società del secolo: quello dei corrispondenti.

Tanto per intenderci cominciò a invalere l'uso per i nomi più grandi, e anche per quelli di più piccolo calibro, di intrattenere una fitta corrispondenza. I grandi capoccioni del tempo utilizzavano tale mezzo per propagandare l'annuncio delle loro scoperte.

Uno dei più illustri e prolifici amici di penna dei grandi della prima metà del secolo (era uno dei più ricercati, una specie di agente letterario ante litteram) fu tal padre Marino Mersenne, che formò la sua cultura al collegio della Flèche e pare fu compagno di corso di Cartesio. Nel 1611 vestì l'abito dei frati minimi di S. Francesco De Paola. Personalità piuttosto nota e dotata di influenza su diverse corti europee assunse il compito di arbitro, non sempre imparziale, nelle dispute scientifiche del tempo. È noto, soprattutto, per due cose: l'aver tradotto, benché ecclesiastico, in francese i «Dialoghi sui massimi sistemi» di Galileo, proprio nel periodo più scottante della sua condanna, e per aver tenuto, come dicevamo, una fitta corrispondenza, tutta ben conservata con diverse personalità scientifiche dell'epoca.

Quello che però a noi interessa del nostro buon frate è legato ai suoi «numeri»; questi sono numeri della forma:

$2^p - 1$

dove p è un numero intero.

I numeri di tal fatta sono detti numeri di Mersenne e, nel caso p sia primo, rappresentano quasi sempre numeri primi. Abbiamo detto quasi, perché se è vero che per 2, 3, 5 e 7, da sostituire a p, la cosa è verificata, per 11 il numero non è primo: infatti

$2^{11} - 1 = 2047$

che, con il programma precedentemente presentato dà come fattori 23 ed 89.

Nella storia dei numeri il più grande dei numeri primi conosciuti è sempre stato un numero di Mersenne (tranne un breve periodo intorno al 1951). Fino al 1983 il più elevato numero prima conosciuto era:

$2^{44497} - 1$

scoperto e pubblicato da David Slowinski nell'aprile del 1979. In quella data, lo stesso Slowinski ne trovò uno enormemente più grande, il

$2^{86243} - 1$

```
'        Programma per la ricerca dei fattori primi di un numero
'
'                  versione 1.0
'
'        Il programma calcola i fattori primi di un numero
'
    CLS
    WIDTH 60
    PRINT " Determinazione dei fattori primi di un numero "
    PRINT
    PRINT "              ------------ "
    PRINT
'
2
    INPUT " Indicare il numero che si intende testare  ", numero
    IF numero < 7 OR numero <> INT(numero) THEN PRINT "attento,un numero valido " GOTO 1
'
3
    PRINT
    PRINT " I fattori primi del numero " ; numero ; " sono : "
'
4
    uno = SQR(numero) : due=uno+1
'
    IF numero = 2 * INT (numero/2) THEN tre = 2 : GOSUB 10
    IF numero = 3 * INT (numero/3) THEN tre = 3 : GOSUB 10
'
    FOR loop = 5 TO uno STEP 6
      IF numero = loop * INT(numero/loop) THEN tre = loop : GOSUB 10
      IF numero = (loop+2) * INT(numero/(loop+2)) THEN tre = loop+2 : GOSUB 10
      IF loop > due THEN loop = uno
    NEXT loop
'
    IF numero > 1 THEN PRINT numero ,
'
'
'
'
5
    PRINT
    INPUT " vuoi ricominciare (S/N) " , risposta$
    IF UCASE$(risposta$) = "S" THEN 2
'
6
    END
'
    ******************
'
10
    ' routine per la determinazione dei fattori primi
'
11
    PRINT tre,  numero = INT(numero/tre)
    IF numero > tre*INT(numero/tre) THEN 12
    GOTO 11
12
    due = SQR(numero)+1
'
    RETURN
```

Figura 3 - Programma per la ricerca dei fattori primi di un numero.

```
                    versione 1.0

          Il programma verifica, in chiave probabilistica
                la primarità di un numero


CLS
RANDOMIZE TIMER

WIDTH 60
PRINT " Determinazione della probabile primarità di un numero "
PRINT " col metodo di Fermat-Pomerance"
PRINT
PRINT "          ------------"
PRINT

INPUT " Indicare il numero che si intende testare: ", numero
IF numero<7 OR numero<>INT(numero) THEN PRINT"attento,un numero valido ":GOTO 1

uno = 0 : due = numero-1: flag1=0

IF due = 2 * INT ( due/2) THEN uno = uno+1 : due = INT ( due/2): GOTO 5

nbase = ABS(INT ( RND * numero - 1)) : IF due < 2 THEN due = 2
PRINT
PRINT " Eseguo il test usando come base" ; nbase
tre=1

IF due <> 2 * INT (due/2) THEN tre = tre*nbase : tre = tre -numero* INT (tre/numero)
nbase=nbase*nbase : nbase = nbase- numero*INT ( nbase/numero) : due = INT(due/2)
IF due = 0 THEN 8
GOTO 7

flag2 = 1
IF tre = 1 OR tre = numero-1 THEN flag2 = 0 : flag3 = 1
IF flag2 AND uno<2 THEN PRINT " Numero non primo ": flag1 = 1
IF flag2 AND uno<2 THEN flag2 = 0 : flag3 = 0: GOTO 10
IF flag2 THEN tre = tre * tre : tre = tre - numero * INT (tre/numero) : uno = uno-1
IF NOT flag2 THEN 9
GOTO 8

IF flag3 <> 1 OR flag1 = 1 THEN PRINT " numero non primo": GOTO 10
PRINT " Il numero è probabilmente primo "
INPUT " desideri testare lo stesso numero per un'altra base (S/N) " , g$

IF UCASE$(g$) = "N" THEN 10 ELSE 4

PRINT
INPUT " vuoi riprovare con un altro numero ( S/N) " , g$
IF UCASE$(g$) = "S" THEN 3

END
```

Figura 4 - Test della primarità di un numero col metodo di Fermat-Pomerance.

composto da ben 25962 cifre decimali (tanto per intenderci questo articolo comprende circa 12000 caratteri). Slowinski usò per la sua ricerca un computer Cray-1; si tratta della più potente macchina commerciale esistente al mondo (è citata, tra l'altro, per la sua rapidità di esecuzione nel Guinness dei primati); con tutto ciò essa impiega, per testare tal numero, più di un'ora e mezza (per la precisione 1 ora 38 minuti e 11 secondi, compreso l'output di risposta sullo schermo).

Come se non bastasse, l'anno appresso fu trovato un numero ancora più grande,

$2^{132049} - 1$

ancora della forma, come si vede, dei numeri di Mersenne. Ma comunque è possibile che chi scrive non sappia di nuove, più grosse (è il caso di dirlo) scoperte in questo campo.

Ovviamente testare un numero di Mersenne col procedimento descritto nei listati appena mostrati è al di fuori di ogni possibilità, anche per il più veloce dei computer. Ma esiste una tecnica diversa, ideata da G.F. Palantier, un matematico lussemburghese, verso la fine del secolo scorso, e successivamente diverse volte perfezionata: essa stabilisce che, per testare un numero di Mersenne della classica forma:

$M = 2^P - 1$

è sufficiente definire la serie:

$U_1 = 4$
$U_2 = (U_1 \cdot U_1 - 2) \text{ MOD } N$
$U_3 = (U_2 \cdot U_2 - 2) \text{ MOD } N$
.....
$U_{p-1} = (U_{p-2} \cdot U_{p-2} - 2) \text{ MOD } N$

N è primo se e solo se U_{p-1} è pari a 0. Ciò vuol dire che, per stabilire se un numero formato Mersenne è primo, è sufficiente eseguire un numero P di operazioni abbastanza banali. Il test di $(2^{132049} - 1)$ è, pertanto, ridotto in forma piuttosto banale, se non fosse per il fatto che si stanno manipolando numeri di più di un centinaio di migliaia di cifre, contro le 10-15 cifre maneggiabili senza approssimazione da un computer (e, ovviamente, qui non sono ammessi arrotondamenti).

Niente paura! esiste una tecnica, geniale e piuttosto semplice, per far manipolare ad un computer, anche il più piccolo home, numeri ben più lunghi di quelli ammissibili: ne parleremo al più presto. Per adesso credeteci sulla parola.

Avendo a disposizione un tempo illimitato è possibile verificare la primarietà di un numero col metodo delle divisioni precedentemente descritto. Ma il tempo è limitato, anche per un computer.

In tempi piuttosto recenti è stato messo a punto un nuovo metodo per la ricerca dei numeri primi, basato su un vecchio teorema del grande Pierre Fermat (ma pare che il vero scopritore sia stato un matematico suo contemporaneo, il gesuita fiammingo Gregorio di S. Vincenzo; Fermat se ne attribuì la paternità, secondo un'abitudine molto diffusa all'epoca, e cui non rifuggì neppure il grande Cartesio); secondo tale teorema, dimostrato, se P è un numero primo e B è un numero compreso tra 1 e P-1 il numero $B^{(p-1)} - 1$ è divisibile per P. Ad esempio, se P = 13 e B = 2, allora $2^{(13-1)} - 1$, vale a dire 2085, è divisibile per 13. Una più ridotta dimostrazione del teorema era comunque già stata offerta dal matematico cinese Pomerance, vissuto nel V secolo A.C., che ne aveva mostrata la validità per B = 2 solamente. Ancora, erroneamente, Pomerance aveva pensato che l'inverso fosse altrettanto valido, vale a dire che, se $2^{(P-1)} - 1$ è divisibile per P allora P è un numero primo. Non sempre ciò accade! Ad esempio 341 è un divisore di $2^{(341-1)} - 1$, ciononostante non è primo.

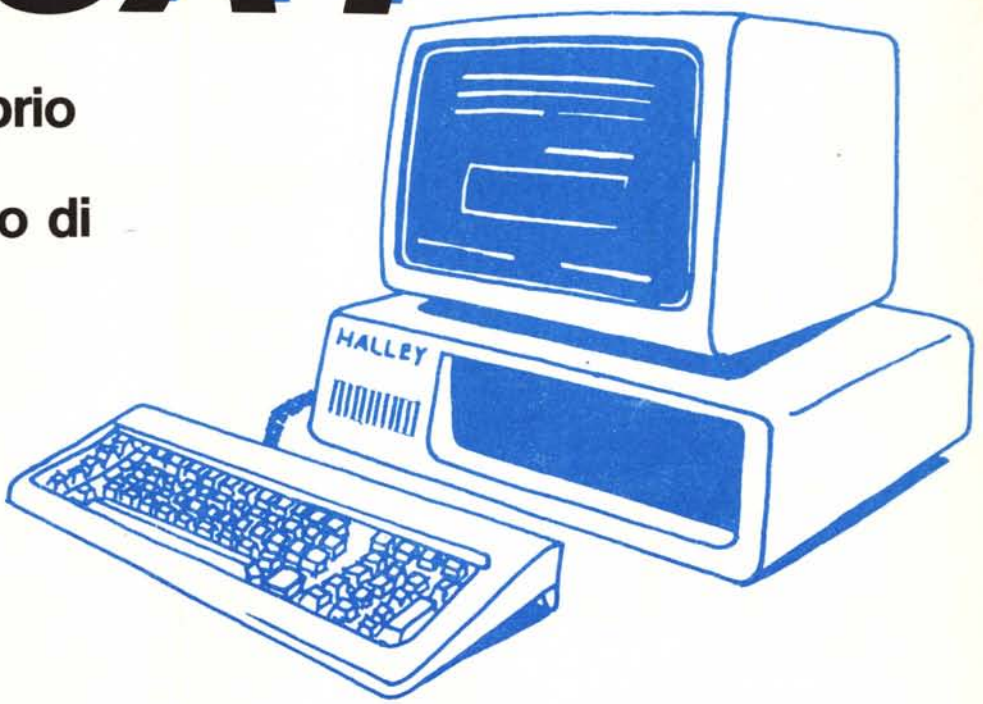
Questi numeri vengono definiti come «Pseudoprimi in base n» dove n è il numero rappresentante la base della potenza stessa. Nonostante, però, Pomerance si sia sbagliato, è pur vero che la quasi totalità di tal messe di numeri è prima.

Il listato della figura 4 serve appunto a testare numeri, di cui si vuol conoscere la primarità, con tale metodo: lo sviluppo è del tutto intuitivo, essendo redatto utilizzando le più elementari routine del MS Basic. L'unico statement parzialmente oscuro, il RANDOMIZE TIMER, non è altro che il normale RANDOMIZE cui è stato assegnato, come seme, il valore del timer interno. Ad esso è deputata la scelta casuale della base. Un numero che risulti pseudoprimo dopo un test con quattro o cinque basi diverse ha molte probabilità di essere primo per davvero! Il programma prevede che dopo ogni prova venga richiesto se si desidera di nuovo riprovare, ma l'introduzione di un banale loop consentirà di effettuare test più lunghi, ad esempio con tre o cinque basi alla volta.

MC

HALLEY... PENSA !

Si, ha pensato proprio a Voi, offrendosi all'incredibile prezzo di L. 1.690.000.



Il computer HALLEY accetta tutto il software sviluppato per il computer IBM PC/XT* come l'MS-DOS 1.1 e 2.0 ecc., l'UCSD-p System, il GW-BASIC, il CP/M-86, il Lotus 1-2-3, il Multiplan, il Woordstar, il VisiOn e tantissimi altri. È un vero IBM PC/XT* compatibile, ha un prezzo assolutamente imbattibile. HALLEY è distribuito dalla CAFCO s.r.l. in tutto il territorio italiano ed è disponibile nelle seguenti versioni:

HALLEY CFC-1000

- CPU 8088-4,77 MHz; coprocessore matematico 8087 (opzionale).
- 256 KB RAM, espandibile a 640 KB on board.
- BIOS (IBM* compatibile).
- 8 K ROM espandibile fino a 48 KB on board.
- Un 5 1/4" floppy disk slim drive da 360 KB DD/DS.
- Scheda standard 6845 per grafica a colori incorporata, 16 KB di memoria video.
- 1 porta parallela per collegamento con stampante.
- Tastiera IBM* compatibile - versione italiana o USA a scelta.
- Monitor 12" monocromatico professionale.

Lire 1.690.000 (I.V.A. esclusa).

HALLEY CFC-2000

Idem come per mod. CFC-1000 con le seguenti varianti:

- Due 5 1/4" floppy disk slim drives da 360 KB DD/DS.

Lire 1.890.000 (I.V.A. esclusa).

HALLEY CFC-2100 Portatile

Idem come per mod. CFC-2000 con monitor 9" monocromatico incorporato.
L. 2.440.000 (I.V.A. esclusa).

HALLEY CFC-6000

Idem come per mod. CFC-2000 con le seguenti varianti:

- 640 KB RAM.
- 1 slim 5 1/4" floppy disk drive da 360 KB DD/DS.
- 1 slim hard disk da 10 MB.

Lire 3.490.000 (I.V.A. esclusa)
Lire 3.990.000 (I.V.A. esclusa)
con hard disk drive da 20 MB.

HALLEY CFC-8000

(IBM* PC/AT compatibile)

- CPU 80286 a 16/24 bit; coprocessore matematico 80287 (opzionale)
- 640 KB RAM espandibile fino a 3 MB.
- 1 floppy disk drive da 1,2 MB DD/DS.
- 1 hard disk drive da 20 MB.
- Scheda grafica/colore incorporata.
- Tastiera IBM* compatibile.
- Monitor 12" monocromatico professionale.
- Possibilità di espansione della memoria di massa fino a 41,2 MB.

Lire 6.990.000 (I.V.A. esclusa).

Distributori:
PIEMONTE: L.P.G. - Corso Allamano Canonico 40/6 - 10136 Torino - Tel. 011/323161-356612.
LOMBARDIA: VIETTI RAPPRESENTANZE - Via Zuccoli 26 - 20125 Milano - Tel. 02/6888437-6889919.
VENETO: EDO SISTEMI s.r.l. - Via Vecchia Valpolicella - 37029 S. Pietro in Cariano (Vr) - Tel. 045/7703677 - Telex 481028.
EMILIA ROMAGNA: VENANZIO BENVENUTI - Via Guelfa 12/11 - 40100 Bologna - Tel. 051/533007.
CAMPANIA: CIPRE s.r.l. - Via E. Gianturco 10 - 80142 Napoli - Tel. 081/267773 - Telex 721253.

CAFCO s.r.l. Via Roggiuzzole 1, 33170 Pordenone, Tel. 0434/550340-550044
Telex 460848 - Telefax 0434/550044

*IBM e IBM PC/XT sono marchi registrati dalla International Business Machines.

Desidero ulteriori informazioni al seguente recapito:

Nome _____
Cognome _____
Indirizzo _____
Telefono _____