

AVVIO ALLA RICORSIVITÀ ATTRAVERSO PROBLEMI *

Consolato PELLEGRINO

Dipartimento di MATEMATICA - Università di MODENA

Dopo aver accennato alle finalità del GREM (Gruppo di Ricerca sulla Educazione Matematica - Modena) descriviamo gli obiettivi (§1) e le caratteristiche (§2) delle sue proposte relative all'introduzione dell'Informatica nella didattica della Matematica. Infine (§3) illustriamo, per sommi capi, una proposta di avvio alla ricorsività per allievi di 11-14 anni (cfr. Pellegrino-Garuti 1990) che può offrire spunti interessanti anche ad insegnanti di Scuola Secondaria Superiore.

1. - LA NOSTRA IDENTITÀ

Dal 1985 opera, presso il Dipartimento di Matematica dell'Università di Modena, il GREM (Gruppo di Ricerca sulla Educazione Matematica). Il GREM, cui aderiscono vari insegnanti di Scuola Secondaria, svolge attività di ricerca finalizzata al miglioramento dell'insegnamento e dell'apprendimento della Matematica per allievi di 11-16 anni con particolare attenzione sia alle innovazioni curricolari che alla gestione del lavoro in classe da parte dell'insegnante (cfr. Malara-Pellegrino 1990). Più precisamente tale ricerca punta alla:

- a) valorizzazione degli aspetti formativi e culturali della Matematica attraverso la trasmissione delle idee e dei metodi che sono alla base del *fare Matematica*;
- b) introduzione *attiva* dell'Informatica nella didattica della Matematica attraverso un uso del computer come *macchina per pensare*.

Per quanto riguarda più specificatamente il precedente punto b) il GREM mira a:

- realizzare l'inserimento dell'Informatica nella didattica della Matematica attraverso esperienze che - basate su *attività o problemi opportunamente scelti* e con l'ausilio di *metodologie appositamente elaborate* - permettono un graduale trasferimento di competenze informatiche dall'insegnante agli allievi *saltando la fase di alfabetizzazione*;
- individuare difficoltà di apprendimento e strategie produttive messe in atto dagli allievi attraverso l'osservazione del loro comportamento durante attività che preludono all'uso del calcolatore o ne facciano esplicitamente uso.

In altri termini il GREM mira ad una introduzione dell'Informatica nella Scuola più come fatto culturale che come semplice avvio all'uso dei calcolatori o all'apprendimento di

* Lavoro con il contributo del CNR (contratto n°90.01147.CT01)

usata a sproposito da altri, potesse offrire agli allievi valide occasioni di riflessione e approfondimento. L'esperienza è stata progettata con cura e seguita con molta attenzione: noi stessi avevamo qualche timore circa la sua realizzabilità in tutte le sue parti. Essa si è svolta, senza grossi problemi, utilizzando come unico prerequisito il concetto di sistema di numerazione in base K.

Considerato l'interesse destato in allievi così giovani e le valenze didattiche offerte da questa attività noi riteniamo che, con semplici adattamenti (quali ad esempio, se necessario, l'impiego del PASCAL al posto del LOGO) ed eventuali ampliamenti, essa possa essere proficuamente proposta ad allievi di classi non sperimentali del biennio della Scuola Secondaria Superiore. A questo fine riportiamo in appendice le versioni in PASCAL delle procedure realizzate.

L'esperienza si può essenzialmente dividere in una parte di avvio ed in una di approfondimento e, come previsto, ha offerto ai ragazzi un ricco ventaglio di occasioni di esplorazione, riflessione e confronto.

Nella prima parte (avvio) si è affrontata una discussione finalizzata ad evidenziare il concetto di successivo dei numeri naturali, si è costruita poi con gli allievi la procedura SUCCESSIVO (fig. 1.d) che permette di simulare il passaggio da un numero al suo successivo in un contachilometri. Si è poi utilizzata la procedura:

```
PER KONTATORE :NUM
  STAMPA :NUM
  KONTATORE SUCCESSIVO :NUM
FINE
```

che mediante SUCCESSIVO permette di simulare il funzionamento del contachilometri (ossia permette di visualizzare il succedersi dei numeri come avviene in un contachilometri di un'auto in movimento) e si sono fatte osservazioni sulla natura ricorsiva di queste due procedure (in particolare si è fatto notare che KONTATORE una volta lanciata, senza un intervento esterno non si arresterebbe mai).

Nella seconda parte (approfondimento) si sono adattate SUCCESSIVO e KONTATORE in modo da simulare il contachilometri in base otto. Successivamente si sono portati i ragazzi a costruire le seguenti procedure:

```
PER TRASFORMATO :NUM
  SE :NUM < 8 RIPORTA :NUM
  RIPORTA (8 * TRASFORMATO / 8) + (NUM / 8)
FINE
```

```
PER INVERSO :NUM
  SE :NUM < 8 RIPORTA :NUM
  PAROLA (INVERSO / 8) (RESTO / 8)
FINE
```

che consentono rispettivamente di passare dalla rappresentazione di un numero in base otto alla sua rappresentazione in base dieci e viceversa.

Fig. 1.- Costruzione della procedura SUCCESSIVO: (a) tabella riassuntiva dei fatti "osservati"; (b) traduzione in LOGO dei fatti osservati; (c) descrizione delle primitive impiegate; (d) inserendo in SUCCESSIVO una opportuna istruzione, come indicato dalla freccia, si ottiene una nuova procedura che funziona anche nel caso limite che il numero sia costituito da tutti nove.

<p>(a)</p> <p>Per fare il successivo si un numero</p> <p>se l'ultima cifra del numero è 0 riporta il numero che ottieni facendo seguire 1 al numero dato privato dell'ultima cifra</p> <p>se l'ultima cifra del numero è 1 riporta il numero che ottieni facendo seguire 2 al numero dato privato dell'ultima cifra</p> <p style="text-align: center;">:</p> <p style="text-align: center;">:</p> <p>se l'ultima cifra del numero è 9 riporta il numero che ottieni facendo seguire 0 al successivo del numero dato privato dell'ultima cifra</p> <p>fine</p>	<p>(b)</p> <p>PER SUCCESSIVO :NUM</p> <p>SE ULT :NUM = 8 RIPORTA PAROLA SUCCESSIVO MENULT :NUM 1</p> <p>SE ULT :NUM = 1 RIPORTA PAROLA SUCCESSIVO MENULT :NUM 2</p> <p style="text-align: center;">:</p> <p style="text-align: center;">:</p> <p>SE ULT :NUM = 9 RIPORTA PAROLA SUCCESSIVO MENULT :NUM 0</p> <p>FINE</p>
<p>(c)</p> <p>RIPORTA : applicata ad un <i>oggetto</i> LOGO (numero, parola o lista) interrompe l'esecuzione della procedura in corso e restituisce quell'<i>oggetto</i> (serve per definire funzioni).</p> <p>PAROLA : applicata a due sequenze di caratteri (numeri o parole) restituisce la sequenza che si ottiene «concatenando» la prima sequenza con la seconda. Es.: PAROLA 12 34 restituisce 1234 mentre PAROLA "BUON "GIORNO restituisce BUONGIORNO.</p> <p>ULT : applicata ad un oggetto LOGO restituisce l'ultima componente di quell'oggetto. Es.: ULT 123 restituisce 3, ULT "FINE restituisce E, mentre ULT [IL CANE ABBAIA] restituisce ABBAIA.</p> <p>MENULT : applicata ad un oggetto LOGO restituisce quell'oggetto privato dell'ultima componente. Es. MENULT 123 restituisce 12, MENULT "FINE restituisce FIN, mentre MENULT [IL CANE ABBAIA] restituisce IL CANE.</p>	
<p>(d)</p> <p style="text-align: center;">PER SUCCESSIVO :NUM</p> <p style="text-align: center;">←————— SE :NUM = "9 RIPORTA "0</p> <p>SE ULT :NUM = 0 RIPORTA PAROLA MENULT :NUM 1</p> <p>SE ULT :NUM = 1 RIPORTA PAROLA MENULT :NUM 2</p> <p>SE ULT :NUM = 2 RIPORTA PAROLA MENULT :NUM 3</p> <p>SE ULT :NUM = 3 RIPORTA PAROLA MENULT :NUM 4</p> <p>SE ULT :NUM = 4 RIPORTA PAROLA MENULT :NUM 5</p> <p>SE ULT :NUM = 5 RIPORTA PAROLA MENULT :NUM 6</p> <p>SE ULT :NUM = 6 RIPORTA PAROLA MENULT :NUM 7</p> <p>(*) SE ULT :NUM = 7 RIPORTA PAROLA MENULT :NUM 8</p> <p>(*) SE ULT :NUM = 8 RIPORTA PAROLA MENULT :NUM 9</p> <p>SE ULT :NUM = 9 RIPORTA PAROLA SUCCESSIVO MENULT :NUM 0</p> <p>FINE</p>	

Più precisamente, tranne che per KONTATORE, in conformità alla nostra metodologia, il lavoro di progettazione-realizzazione-verifica-studio di ciascuna delle procedure SUCCESSIVO, TRASFORMATO e INVERSO si è svolto attraverso le seguenti fasi che con allievi più grandi si possono conservare nella sostanza adattando convenientemente la forma:

- a) presentazione del problema mediante schede di lavoro e/o discussione collettiva (in questa fase si pone particolare attenzione a catturare e stimolare l'interesse degli allievi);
- b) analisi del problema mediante opportune schede e discussione collettiva;
- c) stesura di un conveniente elenco di fatti osservati attraverso la sistematizzazione e la riorganizzazione del lavoro compiuto nella fase precedente (vedi ad esempio fig. 1.a);
- d) traduzione in un adeguato linguaggio di programmazione (in questo caso il LOGO: vedi ad esempio fig. 1.b) dei fatti osservati (soprattutto all'inizio quando gli allievi non hanno alcuna conoscenza di tipo informatico è l'insegnante che, *proponendosi come interprete tra gli allievi ed il computer*, realizza detta traduzione: questo artificio consente di ridurre la fase di alfabetizzazione informatica a poco più che qualche cenno sulle primitive necessarie (vedi ad esempio fig. 1.c), inoltre questa attività consente di posticipare e ridurre al minimo gli aspetti teorici e rende facile l'avvio alla ricorsività);
- e) prove alla macchina ed eventuale rilevazione di errori (generalmente guidate mediante *schede di collaudo* in cui si invitano gli allievi a confrontare le risposte da loro attese con quelle effettivamente fornite dal computer - vedi ad esempio fig. 2);
- f) analisi del funzionamento della procedura effettuato mediante una conveniente *tabella di simulazione*, detta del *come funziona* (serve a scoprire, quando è il caso, dove e perché una procedura porta a risposte errate inoltre questa attività porta gli allievi a *vedere* i singoli passi compiuti dalla macchina ed ad entrare così nel meccanismo della ricorsività - vedi ad esempio fig. 3);
- g) correzione degli errori scoperti attraverso la sistemazione dei casi limite, dei difetti dovuti alla limitatezza dei numeri di macchina, ecc.;
- h) discussione ed approfondimenti (questa fase che può comprendere l'eventuale raffinamento della procedura realizzata consente agli allievi di vedere la genesi di regole matematiche - vedi ad esempio fig. 4).

La prima procedura (SUCCESSIVO) ha richiesto parecchio tempo per essere realizzata e fatta propria dagli allievi, le altre invece sono state costruite via via con maggiore facilità. Ricordiamo che per la realizzazione di SUCCESSIVO l'insegnante ha svolto il ruolo di "interprete" assumendosi interamente il compito di tradurre in LOGO l'algoritmo elaborato insieme ai ragazzi ma che con il progredire dell'esperienza c'è stato un graduale trasferimento di competenze dall'una agli altri: alla fine questi ultimi hanno acquisito una certa familiarità con questo procedimento tanto che le loro soluzioni dei problemi proposti erano praticamente procedure LOGO.

Ci sembra importante sottolineare, per ciò che riguarda la metodologia, che il ricorso alle tabelle del *come funziona* si è rivelata una scelta didattica felice in quanto esse hanno reso *trasparente* agli allievi sia il *meccanismo nascosto* delle procedure realizzate, sia il funzionamento di ciascuna di esse. La spiegazione del meccanismo nascosto ha messo in luce la natura ricorsiva di dette procedure ossia ha permesso di evidenziare il fatto che: *le procedure ricorsive risolvono i problemi riconducendoli ad altri ad esso analoghi che diventano via via più semplici sino al punto da poter essere risolti immediatamente* (cfr. Roberts tr.it. 1987, pp. 17-18). Invece il funzionamento di ciascuna di esse ha permesso

Fig. 2. - Prove alla macchina e rilevazione di eventuali errori di SUCCESSIVO

(a) *Scheda di collaudo*

La procedura SUCCESSIVO, se tutto va bene, dovrebbe rispecchiare il funzionamento del contachilometri nel passaggio da un numero al suo successivo. Scrivi accanto a ciascuno dei seguenti comandi quale dovrebbe essere, secondo te, la risposta del computer.

COMANDI	RISPOSTE TUE	RISP. COMPUTER
a) STAMPA SUCCESSIVO 8785	<u>8786</u>	<u>8786</u>
b) STAMPA SUCCESSIVO 326	<u>327</u>	<u>327</u>
c) STAMPA SUCCESSIVO 7769	<u>7770</u>	<u>7770</u>
d) STAMPA SUCCESSIVO 779	<u>780</u>	<u>780</u>
e) STAMPA SUCCESSIVO 10	<u>11</u>	<u>11</u>
f) STAMPA SUCCESSIVO 0326	<u>0327</u>	<u>0327</u>
g) STAMPA SUCCESSIVO 0010	<u>0011</u>	<u>11</u>
h) STAMPA SUCCESSIVO 0000	<u>0001</u>	<u>1</u>
i) STAMPA SUCCESSIVO 1234567890123	<u>1234567890124</u>	<u>123456 E 12</u>
l) STAMPA SUCCESSIVO 999	<u>1000</u>	<u>M. E.</u>
m) STAMPA SUCCESSIVO 9	<u>10</u>	<u>M. E.</u>

Dopo aver trasferito SUCCESSIVO nella macchina scrivi le risposte effettivamente date dal computer.
Qualche risultato ti sorprende perché?

(b) *Risultati del collaudo*

La traduzione LOGO dei fatti evidenziati (fig. 1.b), nella sostanza, sia pure con qualche inconveniente, è una procedura che consente al computer di simulare il contachilometri nel passaggio da un numero al suo successivo.

(c) *Inconvenienti riscontrati*

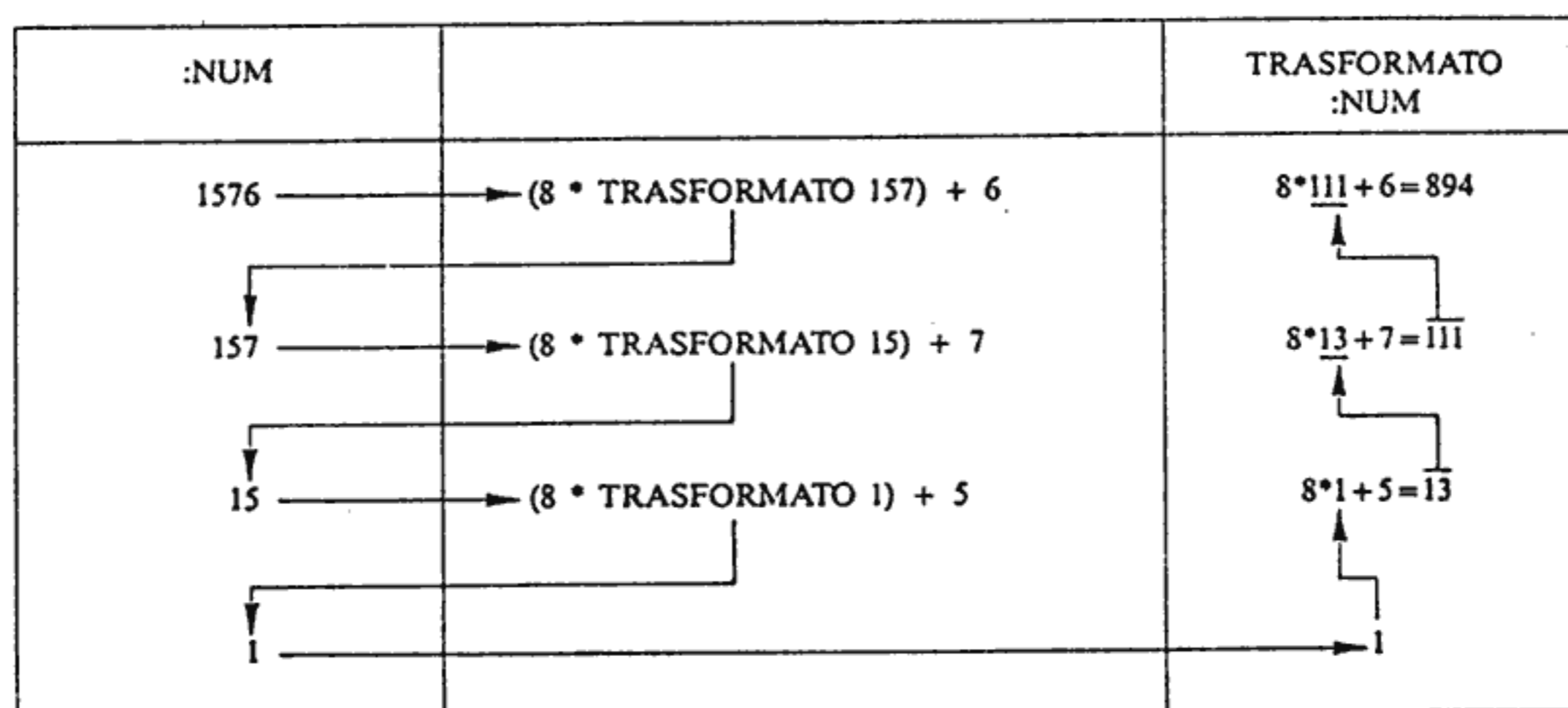
- 1) se il numero inizia con uno o più zeri, essi vengono eliminati nella risposta del computer;
- 2) se il numero è troppo "grande", esso viene espresso con notazione esponenziale (i numeri gestibili dalla macchina sono limitati);
- 3) nel caso limite che il numero sia composto da tutti 9, il computer dà un messaggio d'errore di non facile interpretazione.

(d) *Rimedi*

- Per eliminare (1) e (2) basta aggiungere le virgolette (") davanti al numero (es.: "00327 invece di 00327). E' un esempio di correzione di errore mediante il ricorso alla *strutturazione dei dati*: le virgolette dicono al LOGO di trattare NUM come stringa, ossia come una sequenza di caratteri.
- Per eliminare (3) basta inserire, come prima riga di SUCCESSIVO, l'istruzione:
SE :NUM = "9 RIPORTA "0

E' un esempio di correzione di errore mediante l'aggiunta di una apposita istruzione che consenta di trattare il caso limite in cui NUM è composto da tutti 9.

Fig. 3. - Come funziona TRASFORMATO



anche di evidenziare il fatto che con l'analisi di una procedura ricorsiva si può, anche se a volte con un certo sforzo, costruire una procedura "diretta" di soluzione ossia che specifica direttamente la sequenza di passi elementari senza mai lasciare niente in sospeso o dover tornare indietro (cfr. Luccio 1984, p. 63). Nel caso in esame questa analisi ha consentito agli allievi di vedere la genesi di alcune regole matematiche quali quella derivante dall'analisi di TRASFORMATO che, come illustrato in fig. 4, ricalca il noto schema di Ruffini e consente di passare dalla rappresentazione di un numero in base otto a quella in base dieci.

Un altro aspetto da sottolineare è che durante le prove alla macchina delle procedure realizzate gli allievi hanno potuto effettuare significative attività di scoperta e correzione di errori (fig. 2) rendendosi così conto che spesso la questione da porsi, riguardo a un programma, non è se è giusto o sbagliato, ma se lo si può sistemare (cfr. Papert tr. it. 1980, p. 29).

Con questa esperienza abbiamo avuto modo di verificare la ricchezza di spunti di lavoro e di riflessione che situazioni di laboratorio come questa offrono agli allievi: senza l'uso del computer molte di queste attività sarebbero impensabili o avrebbero scarsa incisività.

Inoltre, quando gli allievi hanno acquistato un minimo di autonomia, al fine di verificare il grado di interiorizzazione del lavoro svolto e di esercitarli alla lettura ed alla comprensione di procedure elaborate da altri, sono stati proposti su scheda brevi listati di procedure con la consegna di prevederne e di descriverne gli effetti sui dati di ingresso.

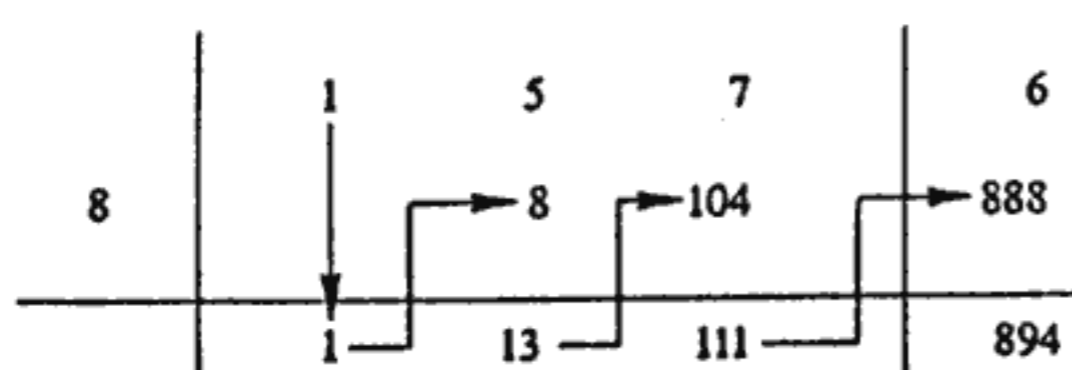


Fig.4. - Passaggio dalla rappresentazione in base otto a quella in base dieci di 1576_8

BIBLIOGRAFIA

- IADEROSAR., 1990, Un'esperienza didattica di interazione tra allievi ed elaboratore per la risoluzione di problemi aritmetici, in corso di stampa su Atti Conv. Naz. *Informatica e Didattica*, Salerno 28-30 settembre 1990.
- LUCCIO F., 1984, Spunti di algoritmica concreta, *Bollettino UMI* (6), 3-A, pp. 57-80.
- MALARA N.A., PELLEGRINO C., 1990, "1985-1990": Cinque anni di attività del GREM, contributo al XIV Conv. Naz. UMI-CIIM *La prima Educazione Matematica*, Cala Gonone (NU) 25-27 ottobre 1990.
- PAPERT S., 1980 (tr. it. 1984), *Mindstorms - Bambini, computers e creatività*, Ed. Emme, Milano.
- PELLEGRINO C., GARUTI R., 1989, Dalla scoperta di regolarità numeriche all'avvio all'Informatica: descrizione di una esperienza didattica, *L'Educazione Matematica*, IV, n.1, pp. 69-81.
- PELLEGRINO C., GARUTI R., 1990, Dall'avvio alla ricorsività ai cambiamenti di base nei sistemi di numerazione attraverso la simulazione in LOGO del contachilometri: descrizione di un'esperienza realizzata in una scuola media, *La Matematica e la sua Didattica*, IV, n.1, pp. 19-30.
- POLYA G., 1945 (tr. it. 1967), *Come risolvere i problemi di Matematica, Logica ed euristica nel metodo matematico*, Feltrinelli, Milano, 1967.
- ROBERTS E. S., 1986 (tr. it. 1987), *Il pensiero ricorsivo*, Franco Angeli Ed., Milano.
- SAWYER W.W., 1964-1970 (tr. it. 1974-1978), *Guida all'insegnamento della Matematica*, 2 voll., Boringhieri, Torino.

APPENDICE: Versione PASCAL delle procedure realizzate

```

program dr;
{SS-}

type stringa=string[10];

{-----}
function ult (x:stringa) : char;
begin
    ult :=x [length (x)]
end; {ult}
{-----}
function menult (x:stringa) : stringa;
begin
    menult :=copy (x,1,length (x)-1)
end; {menult}
{-----}
function success (num:stringa) : stringa;
begin
    success :=num;
    if num='9' then begin success:= '0' ; exit; end;
    if ult (num)='0' then begin success:= menult(num)+'1' ; exit ; end;
    if ult (num)='1' then begin success:= menult(num)+'2' ; exit ; end;
    if ult (num)='2' then begin success:= menult(num)+'3' ; exit ; end;
    if ult (num)='3' then begin success:= menult(num)+'4' ; exit ; end;
    if ult (num)='4' then begin success:= menult(num)+'5' ; exit ; end;
    if ult (num)='5' then begin success:= menult(num)+'6' ; exit ; end;
    if ult (num)='6' then begin success:= menult(num)+'7' ; exit ; end;
    if ult (num)='7' then begin success:= menult(num)+'8' ; exit ; end;
    if ult (num)='8' then begin success:= menult(num)+'9' ; exit ; end;
    if ult (num)='9' then
        begin
            success:=success(menult(num))+'0' ;
            exit
        end;
end; {success}
{-----}
procedure kontatore(var num:stringa);
begin
    writeln(num);
    num:=(success(num));
    kontatore(num);
end; {kontatore}
{-----}

```

(continua)

APPENDICE: Versione PASCAL delle procedure realizzate (continuazione)

```

function trasformato(num:stringa) : stringa;
var n,n1,n2,codice:integer;
begin
    trasformato:='';
    val(num,n,codice);
    if codice>0 then begin writeln('cifra non corretta'); exit end;
    if n<8 then
        trasformato:=num
    else
        begin
            val ( trasformato(menuult(num)) , n1 , codice ) ;
            if codice>0 then begin writeln('cifra non corretta'); exit end;
            val( ult(num) , n2 , codice ) ;
            if codice>0 then begin writeln('cifra non corretta'); exit end;
            if n2>7 then begin writeln('cifra non corretta'); exit end;
            str(8*n1+n2,num) ;
        end;
end; {trasformato}
{-----}
function inverso(num:stringa) : stringa;
var n,codice:integer;
    num1,num2:stringa;
begin
    inverso:='';
    val(num,n,codice) ;
    if codice>0 then begin writeln('cifra non corretta'); exit end;
    if n<8 then
        inverso:=num
    else
        begin
            str (n div 8,num1) ;
            str (n mod 8,num2) ;
            inverso:=inverso(num1)+num2;
        end;
end; {inverso}
{-----}
{
    PROGRAMMA DI PROVA
}

var num:stringa;

begin
    while true do
        begin
            readln(num) ;
            writeln(trasformato(num));
            writeln(trasformato(inverso(num)));
            writeln;
        end;
end.

```