

Roulette

Nel programma presentato in questo numero ci occuperemo della simulazione del famosissimo gioco della Roulette: dalla nostra TI (58, 58C, 59) potremo ottenere una sequenza di numeri interi, casuali, compresi tra 0 e 36; la nostra fedele calcolatrice, nei panni di un integerrimo croupier, ci fornirà anche le indicazioni di rito: "2 Noir Pair Manque".

L'algoritmo

Il nostro problema è tutt'altro che difficile, dato che già esiste nel modulo Master (in dotazione alle TI) un sottoprogramma di generazione di numeri casuali (Pgm 15 SBR D.MS) e dato che è semplice, una volta estratto un numero casuale, decidere se è "Pair" o "Impair" (pari o dispari) e "Manque" o "Passe" (rispettivamente compreso tra 1 e 18 o tra 19 e 36).

Una lieve difficoltà comporta invece il poter decidere se il numero estratto è "Rouge" o "Noir" (rosso o nero), dato che questi colori sono distribuiti a priori secondo criteri probabilistici.

Osservando la fig. 1, che rappresenta una parte del tabellone della roulette e in cui i numeri "Neri" sono preceduti e seguiti da un punto, si nota che questa distribuzione di colori è regolare solo entro certi intervalli.

Il problema è avere un metodo semplice per decidere, partendo dal numero, qual è il suo colore.

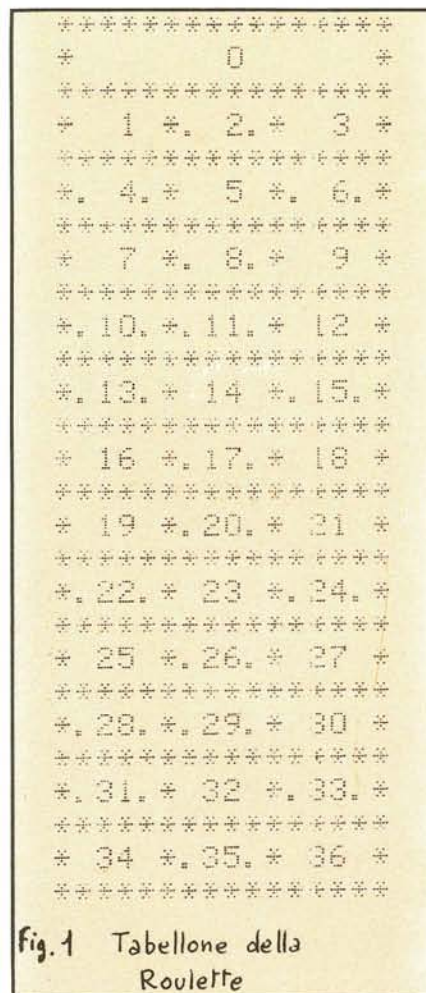
La prima idea, subito scartata (il perché sarà subito evidente); è quella di avere un vettore a 36 componenti, diciamo le memorie da 1 a 36, il cui contenuto numerico può fornirci l'indicazione del colore: al limite si potrebbe riempire ogni elemento di questo vettore con la "codifica" delle scritte Rouge et Noir ed in questo modo, una volta generato un numero casuale tra 0 e 36 si può sfruttare l'indirizzamento indiretto per andare a rilevare il codice da mandare in stampa. Il tutto avviene velocemente in quanto si tratta di semplici operazioni in memoria ed una stampa, però richiede un'occupazione di memoria non conveniente per i modelli TI-58. Inoltre bisogna riempire queste memorie con i due codici e questo già porta via un po' di tempo se effettuato manualmente (con il rischio di compiere errori) oppure porta via ulteriore spazio di memoria di programma se effettuato automaticamente.

Abbandoniamo perciò questa idea anche perché osservando meglio il tabellone (fig. 1) ci si accorge facilmente che la combinazione di colori dei primi 18 numeri si ripete pari pari negli ultimi 18 numeri. Ma non è finito: nell'ambito di ognuno dei due sottoinsiemi si vede che i neri sono i numeri pari fino al decimo incluso e poi i dispari i successivi.

Fatte queste considerazioni, andiamo ad analizzare il flow-chart.

Analisi del programma

Al primo colpo d'occhio il diagramma di flusso appare in un certo punto "non strutturato" per la presenza di tre test che quasi si "intersecano": vedremo in seguito



come ci si comporterà in quel caso.

Generato un numero casuale si fa subito il test se questo è 0 nel qual caso si ha una stampa particolare, dato che particolare è il comportamento del banco nel caso di uscita dello 0 nella roulette vera e propria.

Successivamente si ha il test se il numero è pari o dispari con relativa accensione o spegnimento del flag 0. A questo scopo è utilizzata una caratteristica poco nota della funzione INV, che può essere separata dalla funzione da invertire con un'etichetta.

Andiamo perciò ad osservare i passi 22 e seguenti del programma: si divide per 2 il numero estratto e si prende la parte decimale del risultato. Se questo è zero il numero è pari, altrimenti è dispari; effettuato il test di uguaglianza con 0 (passo 27) in caso di esito positivo (passo 28) si va all'etichetta "Stflg" (LBL STF ai passi 30-31) dopo di che si setta il flag 0 (STF 00 ai passi 32-33) mentre in caso di confronto negativo il programma, dal passo 27, ignora il salto (passo 28) ed esegue l'istruzione del passo 29 che contiene appunto INV.

Nei passi successivi si trova LBL STF, che non influisce sul comportamento dell'INV, e quindi la funzione da invertire, STF 00, ai passi 32 e 33. In questo caso perciò il flag 0 viene resettato senza spreco inutile di passi di programma.

Successivamente, utilizzando la stessa tecnica, si setterà il flag 1 se il numero estratto è "Manque" (compreso tra 1 e 18) e contemporaneamente, nel caso in cui il numero sia invece "Passe", si provvederà a sottrarre 18 alla quantità M che all'inizio era posta uguale ad N; il numero casuale estratto.

È proprio in virtù della similitudine della distribuzione dei colori nei due insiemi di numeri, che è utile avere questa quantità M.

Infatti successivamente viene effettuato il test se M è minore o uguale a 10 e si sfrutta abilmente l'informazione contenuta nel flag 0 per ottenere la stampa della scritta Rouge o Noir. Ancora una volta si è sfruttata la possibilità di porre un'etichetta tra l'INV e la funzione: in questo caso la funzione è IFF 0, realizzandosi così, in base al test precedente, o il test se il flag 0 è acceso oppure il test se il flag 0 è spento.

In realtà il flow-chart effettivamente ottenuto in questo modo è alquanto differente e rappresentato nella fig. 2, ma evidentemente funziona allo stesso modo.

software SOA

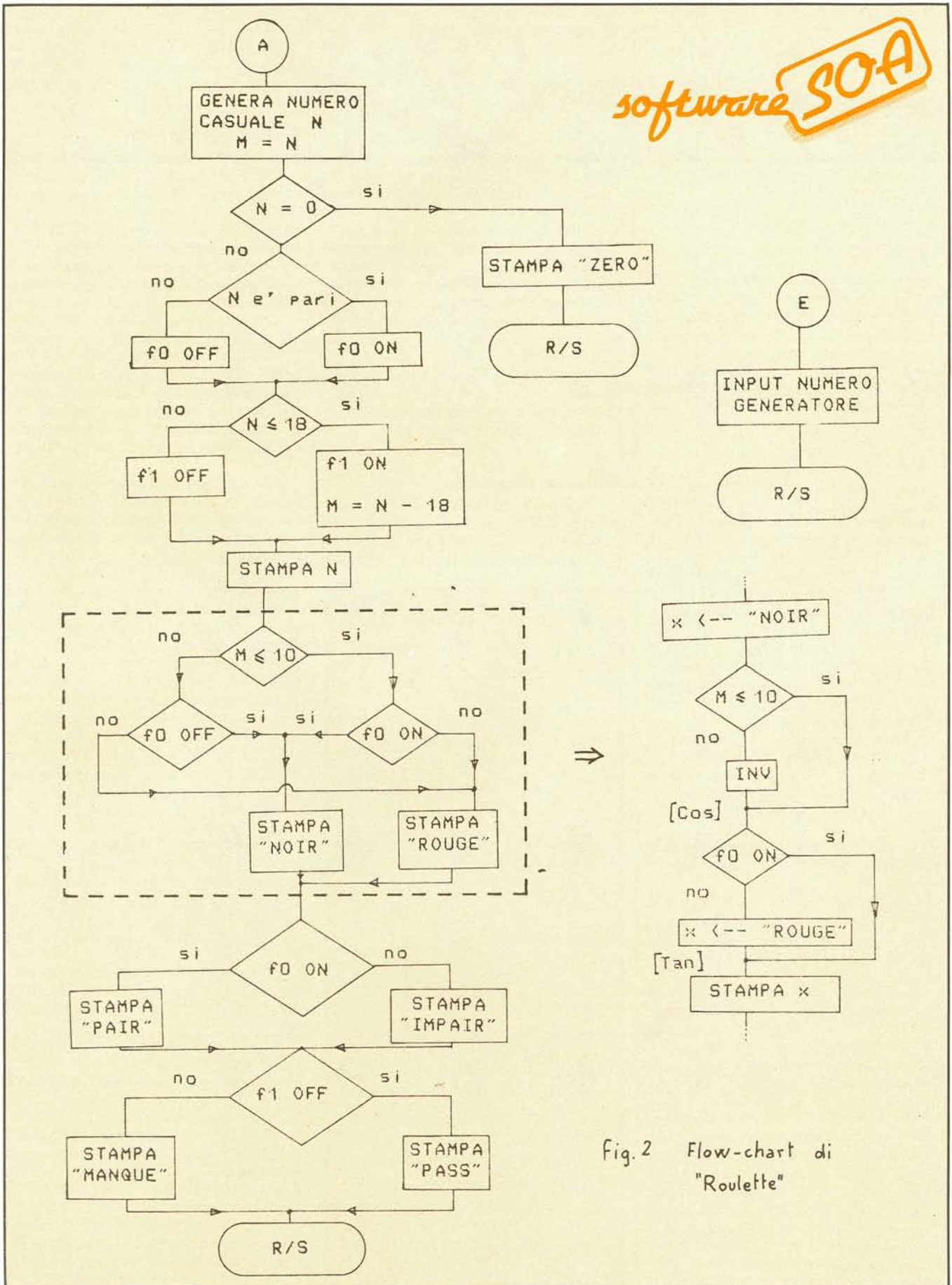


Fig.2 Flow-chart di "Roulette"

000	98	ADV	054	02	2	108	69	DP	162	04	4
001	91	R/S	055	04	4	109	02	02	163	06	6
002	76	LBL	056	03	3	110	76	LBL	164	00	0
003	11	A	057	05	5	111	35	1/X	165	00	0
004	69	DP	058	69	DP	112	69	DP	166	01	1
005	00	00	059	03	03	113	05	05	167	07	7
006	36	PGM	060	43	RCL	114	69	DP	168	00	0
007	15	15	061	01	01	115	00	00	169	00	0
008	71	SBR	062	32	X:T	116	03	3	170	69	DP
009	88	DMS	063	01	1	117	03	3	171	02	02
010	65	*	064	00	0	118	01	1	172	03	3
011	03	3	065	77	GE	119	03	3	173	05	5
012	07	7	066	39	CDS	120	03	3	174	00	0
013	95	=	067	22	INV	121	06	6	175	00	0
014	59	INT	068	76	LBL	122	03	3	176	03	3
015	29	CP	069	39	CDS	123	06	6	177	02	2
016	67	EQ	070	87	IFF	124	01	1	178	00	0
017	67	EQ	071	00	00	125	07	7	179	00	0
018	42	STD	072	30	TAN	126	22	INV	180	05	5
019	00	00	073	03	3	127	87	IFF	181	01	1
020	42	STD	074	05	5	128	01	01	182	69	DP
021	01	01	075	03	3	129	34	FX	183	03	03
022	55	÷	076	02	2	130	03	3	184	69	DP
023	02	2	077	04	4	131	00	0	185	05	05
024	95	=	078	01	1	132	69	DP	186	69	DP
025	22	INV	079	02	2	133	02	02	187	00	00
026	59	INT	080	02	2	134	01	1	188	05	5
027	67	EQ	081	01	1	135	03	3	189	01	1
028	86	STF	082	07	7	136	03	3	190	69	DP
029	22	INV	083	69	DP	137	01	1	191	02	02
030	76	LBL	084	03	03	138	03	3	192	69	DP
031	86	STF	085	76	LBL	139	04	4	193	05	05
032	86	STF	086	30	TAN	140	04	4	194	25	CLR
033	00	00	087	69	DP	141	01	1	195	81	RST
034	43	RCL	088	05	05	142	01	1	196	76	LBL
035	00	00	089	03	3	143	07	7	197	15	E
036	99	PRT	090	03	3	144	76	LBL	198	42	STD
037	32	X:T	091	01	1	145	34	FX	199	09	09
038	01	1	092	03	3	146	69	DP	200	91	R/S
039	08	8	093	02	2	147	03	03	201	00	0
040	77	GE	094	04	4	148	69	DP	202	00	0
041	77	GE	095	03	3	149	05	05	203	00	0
042	22	INV	096	05	5	150	25	CLR			
043	44	SUM	097	00	0	151	81	RST			
044	01	01	098	00	0	152	76	LBL			
045	22	INV	099	69	DP	153	67	EQ	003	11	A
046	76	LBL	100	03	03	154	05	5	031	86	STF
047	77	GE	101	87	IFF	155	01	1	047	77	GE
048	86	STF	102	00	00	156	69	DP	069	39	CDS
049	01	01	103	35	1/X	157	02	02	086	30	TAN
050	03	3	104	02	2	158	69	DP	111	35	1/X
051	01	1	105	04	4	159	05	05	145	34	FX
052	03	3	106	03	3	160	69	DP	153	67	EQ
053	02	2	107	00	0	161	01	01	197	15	E

Inoltre una strutturazione migliore è ottenuta memorizzando prima dei test la parola "Noir" nel registro di stampa (OP 03, passi 50-57), parola che eventualmente viene cancellata e rimpiazzata da "Rouge" nel caso in cui le condizioni dei tre test lo impongano.

Richiamiamo ancora l'attenzione sul fatto che l'insieme di questi tre test è quanto di meno strutturato possa esistere nel campo dei flow-chart e richiede, per l'implementazione, l'uso di parecchi salti. Provatelo a tradurlo ad esempio in BASIC o peggio ancora in Pascal...

Chiudiamo questa digressione che ci ha allontanato dal SOA e ritorniamo al programma.

A questo punto si trova il blocco che fa stampare "Pair" o "Impair" a seconda che il numero sia pari o dispari: analogamente a prima si è sfruttato il fatto che entrambe le parole finiscono con "PAIR" per cui solo nel caso di numero dispari (flag 0 OFF) in fase di stampa si farà precedere questa stringa dalle due lettere "IM".

Infine sfruttando per l'ennesima volta la possibilità di strutturare un blocco del tipo "IFTHENELSE" con un semplice "IFTHEN", si prepara nel visualizzatore il codice di "Passe", che verrà sostituito invece da "Manque" nel caso che il flag 1 è acceso ed in entrambi i casi si passerà per la $Lbl\sqrt{x}$ dopo la quale si memorizza nel registro di stampa il codice presente attualmente nel visualizzatore.

Infine, stampate tutte le informazioni "da croupier", si trova (passo 151) RST

```

26.
  NDIR
  PAIR
  PASSE
6.
  NDIR
  PAIR
  MANQUE
31.
  NDIR
  IMPAIR
  PASSE
32.
  ROUGE
  PAIR
  PASSE
5.
  ROUGE
  IMPAIR
  MANQUE
*
* Z E R O *
*
```

Esempio del programma

L'angolo delle TI

In questa rubrica collegata al Software S.O.A., presenteremo varie notizie, informazioni utili riguardanti caratteristiche poco conosciute o non riportate sui manuali delle tre calcolatrici Texas Instruments TI 58, TI 58C e TI 59.

A parte l'estensione della memoria e altre caratteristiche (memoria costante, lettore di schede), i tre modelli sono fondamentalmente uguali dal punto di vista operativo, avendo un set di istruzioni completamente identico: per questo motivo, a parte eventuali eccezioni che verranno segnalate, tutte le informazioni che compariranno in questa rubrica saranno valide per ognuno dei tre modelli.

Invitiamo perciò i lettori a contribuire con segnalazioni, consigli, richieste.

Indirizzamento ad etichette ed assoluto

La gestione dei salti (condizionato o no) da parte della calcolatrice è differente a seconda che si usino etichette o indirizzamento assoluto.

Nel primo caso la calcolatrice cercherà l'etichetta desiderata partendo in ogni caso dall'istruzione 000, anche se al limite l'etichetta stessa si trova nel passo successivo: ciò comporta che il tempo di ricerca delle etichette dipende dalla loro posizione nella memoria di programma.

Nel secondo caso invece il salto al passo XYZ avverrà in un tempo indipendente dal valore XYZ.

Abbiamo voluto verificare qualitativamente queste differenti durate di elaborazione paragonando la differente durata di un certo numero di cicli di istruzioni contenenti una volta il salto ad etichette e quindi un salto assoluto.

Perciò abbiamo considerato la sequenza

$Lbl\ A\ Dsz\ 0\ A\ R/S$

per l'appunto un loop di decremento unitario del contenuto del registro 00 finché questo si annulla e l'abbiamo posta successivamente a partire dal passo 000, 025, 050, 100, 200, ... 700, 800, 874.

Ogni volta abbiamo inizializzato il registro 00 al valore 20 e abbiamo fatto eseguire il loop misurandone la durata con un orologio.

Si è ottenuto il risultato (qualitativo!) previsto: il tempo di elaborazione cresce linearmente con il crescere dell'indirizzo di partenza del ciclo stesso, in ragione di circa 1 secondo ogni 30 passi di programma.

Abbiamo quindi provato il funzionamento con la seguente sequenza

$Dsz\ 0\ XYZ\ R/S$

dove XYZ è l'indirizzo di partenza della sequenza stessa (000, 025, 050, ecc. come prima); sempre dopo aver posto 20 nel registro 00 (20 STO 00) abbiamo misurato le durate di elaborazione ottenendo un tempo pressappoco identico in ogni caso: 3.5 secondi circa.

Ciò a riprova del fatto che i salti (condizionati e non) assoluti avvengono in tempi praticamente indipendenti dall'indirizzo, così come risulta costante l'accesso ad una memoria RAM in base all'indirizzo della locazione di memoria desiderata.

Ora però bisogna tener conto di una singolare caratteristica dei vari modelli di TI: la differente velocità di elaborazione riscontrabile in vari esemplari di calcolatrici, anche dello stesso tipo.

Per questo motivo i valori riportati prima si riferiscono alla calcolatrice a nostra disposizione, mentre ripetendo la prova su altri modelli si potranno sicuramente ottenere valori differenti. Proponiamo perciò di effettuare l'esperimento comunicandone i risultati.

A questo proposito sarebbe utile indicare, oltre al tipo di calcolatrice, l'anno di fabbricazione.

Sapete da dove si ricava questo dato?

Basta capovolgere la calcolatrice e leggere il numero riportato in alto a destra, subito sopra alla scritta che indica la località in cui la TI è stata "assembled".

Nella nostra tale numero è 4579 che indica la 45-esima settimana dell'anno 1979. Semplice no!?

che azzerava elegantemente tutti i flag e torna al passo 000 dove, dopo un ADV che fa avanzare la carta, si trova l'R/S che ferma l'elaborazione.

L'uso

L'uso del programma è veramente semplice: si imposta un numero generatore e si preme "E"; dopodiché premendo successivamente "A" si ottiene il primo numero estratto. Altri numeri si otterranno indifferentemente premendo "A" o "R/S".

Nell'esempio riportato, che si spiega da sé, si è usato 51 come numero d'innescio.

Memorie usate, ripartizione, flag

Per questo programma vengono usati solo i registri 00 e 01 nonché 07 e 09 dal Pgm 15; la ripartizione è quella iniziale (gli inglesi dicono "default") per tutti e tre i modelli (3 Op 17 per le 58 e 6 Op 17 per la 59), mentre i flag usati sono come visto lo 0 e l'1.

È richiesta infine la stampante dato che il risultato contiene informazioni alfanumeriche.

Signori, fate il vostro gioco...

MC