

mensile  
d'informatica  
e videogames  
L. 8.000

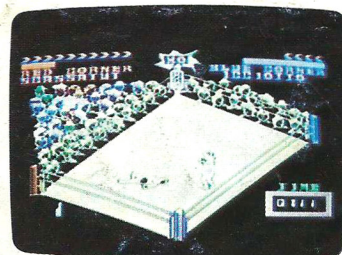
New Video

GAMES

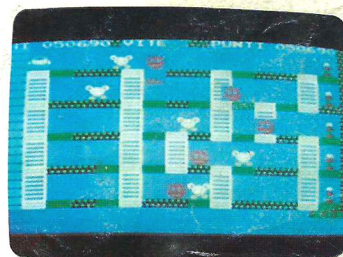
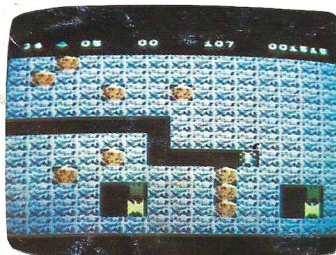
MSX

5

1986  
maggio



- 1 - BOXE
- 2 - LAVANDERIA
- 3 - PUPAZZO DI NEVE
- 4 - CAVERNE
- 5 - AL FUOCO
- 6 - CUBO
- 7 - OTELLO



**VIDEO GIOCHI PER MSX: 32 K RAM**

# New Video

Mensile di informatica  
e video giochi

Anno II  
N. 5 - Maggio '86

EDITORE:  
Editions Fermont s.r.l.  
20121 Milano

REDAZIONE:  
Via Cialdini, 11  
20161 Milano  
Tel. 02/6453775/6

FOTOLITO:  
Claudio Lavezzi  
Via Terruggia, 3  
20162 Milano

STAMPA:  
A.G.E.L. s.r.l.  
Viale dei Kennedy, 92  
20027 Rescaldina

DISTRIBUZIONE:  
MePe  
Via G. Carcano, 32  
20141 Milano

DIRETTORE RESPONSABILE:  
Amilcare Medici

Fotografie di Stefano Monti

**Numeri arretrati:** Ogni numero arretrato £. 8.000 più £. 3.000 di spese postali - Versamento da effettuare sul c/c postale n. 37332202 intestato a EDITIONS FERMONT, Via Cialdini, 11 20161 Milano

## ATTENZIONE

### ISTRUZIONI DI CARICAMENTO

- 1) Dopo aver verificato i collegamenti, computer-video, computer - registratore, computer - rete elettrica, premere il tasto di accensione del computer.
- 2) I giochi girano solo su sistemi MSX: 32 K RAM
- 3) Regolare il tono del registratore sulla posizione massima e il volume a circa 3/4 della sua corsa.
- 4) Digitare: LOAD "CAS:", R
- 5) Battere RETURN (o, in alcuni casi, battere ENTER), quindi premere PLAY.
- 6) All'apparire della schermata con i giochi, spegnere il registratore e computer.
- 7) Riaccendere e digitare: LOAD "CAS:", R
- 8) Battere RETURN.
- 9) Se apparirà il messaggio: DEVICE I/O ERROR, riavvolgere la cassetta, modificare il volume del registratore (alzandolo o abbassandolo leggermente) e ripetere l'operazione.

# Sfruttiamo il nostro MSX

## *“Tutto” sul linguaggio Assembler*

Questo mese amplieremo il discorso iniziato nello scorso numero e diremo cosa è realmente il linguaggio ASSEMBLER. Se l'unico modo di rappresentare il codice macchina fosse quello indicato nell'ultimo numero di questa rubrica per MSX, cioè solo con dei numeri binari, vi sarebbero ben poche persone in grado di scrivere programmi in linguaggio macchina. Risulterebbe molto difficile dare un senso ad un programma che si presenti così:

```
0 0 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
. . . .
```

Possiamo però inventare dei nomi simbolici da associare a ciascuna di queste istruzioni. Il linguaggio ASSEMBLER è appunto una “simbolizzazione mnemonica” del codice macchina, che rende quest'ultimo più facilmente leggibile ed interpretabile.

La vera distinzione tra il linguaggio assembler e il linguaggio macchina consiste nella facilità d'uso e di interpretazione. Questa facilitazione viene però pagata a caro prezzo: il linguaggio assembler, infatti, è più facil-

mente interpretabile dall'uomo ma non può essere letto direttamente dalla CPU.

Vi è comunque un'enorme differenza tra l'assembler ed il BASIC. Infatti, per quest'ultimo le istruzioni basic vengono trasformate in una “serie di istruzioni in codice macchina”, mentre nell'assembler per ogni istruzione in questo linguaggio ne esiste una equivalente in codice macchina e viceversa. Tra i due linguaggi, macchina ed assembler, esiste quindi una corrispondenza “UNO A UNO” che li rende funzionalmente equivalenti.

Per rappresentare le istruzioni il linguaggio assembler utilizza simboli mnemonici (= abbreviazioni). Le istruzioni assembler si presentano così in forma simile a:

```
INC HL
```

Questa scrittura può sembrare incomprensibile, ma se si pensa che “INC” rappresenta l'abbreviazione di “INCrementa” ed HL il nome di una variabile, è sufficiente una sola occhiata per interpretare l'istruzione. La stessa istruzione in codice macchina si sarebbe presentata nel modo seguente:

```
0 0 1 0 0 0 1 1
```

In questo caso è possibile "leggere" l'istruzione riconoscendo il numero che la rappresenta, ma la sua lettura non permette di risalire al significato logico dell'istruzione a meno che non si ricorra ad una tabella di conversione esplicitiva.

Un programma scritto in assembler può comunque essere trasformato facilmente in codice macchina. Questo lavoro può essere fatto direttamente da voi o tramite un apposito programma.

Un programma di conversione di questo tipo viene generalmente chiamato ASSEMBLATORE. L'assemblatore può essere visto come un programma che esegue l'alquanto noioso lavoro di trasformare il vostro programma scritto in assembler, in una sequenza logica di istruzioni in codice macchina direttamente interpretabili dallo Spectrum.

A questo punto avrete capito che esiste già un Assemblatore per il sistema MSX e compatibili, ma deve essere tuttavia notato che l'occupazione di memoria dell'assemblatore è piuttosto elevata (circa 6k) e pertanto il suo utilizzo sui sistemi a 16k risulta piuttosto limitato. Infatti, il monitor per linguaggio macchina dei computer MSX occupa circa 7k di memoria ram e pertanto, dopo aver caricato anche l'assemblatore, rimangono a disposizione dell'utente meno di 4k di memoria. Questi 4k di memoria ram devono contenere sia il codice macchina che il programma in assembler, quindi la quantità di

memoria disponibile si riduce a poche centinaia di bytes liberi.

L'unica alternativa al programma assemblatore consiste nell'effettuare da soli la conversione da linguaggio assembler a codice macchina, usando delle tabelle di conversione.

Questo lavoro può apparire a prima vista frustrante e molto oneroso, ma è di estrema praticità e vi fornisce un valido strumento per capire il modo in cui lavora la CPU del vostro MSX. Per questo vi consigliamo, anche nel caso disponiate di un programma assemblatore, di provare a scrivere in assembler ed a trasformare manualmente in codice macchina almeno qualche piccolo programma, prima di utilizzare il programma di traduzione automatica.

Per questo numero il N.V.G. abbiamo concluso, ma in coda a questo articolo troverete la prima parte delle tabelle di conversione da utilizzare per la traduzione dei programmi da assembler a codice macchina. Nelle tabelle troverete i codici mnemonici dello Z80 ordinate alfabeticamente (colonna MNE-MONIC), il loro corrispondente valore esadecimale (colonna HEX).

Nel prossimo numero di questa rubrica introdurremo i concetti base per la programmazione in linguaggio macchina descrivendo la CPU, il suo schema logico e le sue potenzialità. Arrivederci al prossimo mese!!!

## CONVERSIONE DA DECIMALE A BCD

DECIMALE	BCD	DEC	BCD	DEC	BCD
0	0000	10	00010000	90	10010000
1	0001	11	00010001	91	10010001
2	0010	12	00010010	92	10010010
3	0011	13	00010011	93	10010011
4	0100	14	00010100	94	10010100
5	0101	15	00010101	95	10010101
6	0110	16	00010110	96	10010110
7	0111	17	00010111	97	10010111
8	1000	18	00011000	98	10011000
9	1001	19	00011001	99	10011001

# TABELLA DI CONVERSIONE ESADECIMALE

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	00	000
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	256	4096
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	512	8192
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	768	12288
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	1024	16384
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	1280	20480
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	1536	24576
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	1792	28672
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	2048	32768
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	2304	36864
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	2560	40960
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	2816	45056
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	3072	49152
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	3328	53248
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	3584	57344
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	3840	61440

5		4		3		2		1		0	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15

# TABELLA DI CONVERSIONE ASCII

HEX	MSD	0	1	2	3	4	5	6	7
LSD	BITS	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SPACE	0	@	P	-	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	.	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	--
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	←	o	DEL

## I SIMBOLI ASCII

<b>NUL</b> — Nullo	<b>DLE</b> — Perdita Collegamento Dati
<b>SOH</b> — Inizio della testata	<b>DC</b> — Controllo dispositivo
<b>STX</b> — Inizio del testo	<b>NAK</b> — Riconoscimento negativo
<b>ETX</b> — Fine del Testo	<b>SYN</b> — Sincronismo
<b>EOT</b> — Fine della trasmissione	<b>ETB</b> — Fine del blocco di trasmissione
<b>ENQ</b> — Domanda	<b>CAN</b> — Cancella
<b>ACK</b> — Riconoscimento	<b>EM</b> — Fine del mezzo
<b>BEL</b> — Campana	<b>SUB</b> — Sostituto
<b>BS</b> — Spazio posteriore	<b>ESC</b> — Perdita
<b>HT</b> — Tabulazione orizzontale	<b>FS</b> — Separatore di file
<b>LF</b> — Incremento di riga	<b>GS</b> — Separatore di gruppo
<b>VT</b> — Tabulazione verticale	<b>RS</b> — Separatore di record
<b>FF</b> — Alimentazione del modulo	<b>US</b> — Separatore di unità
<b>CR</b> — Ritorno carrello	<b>SP</b> — Spazio (Blank)
<b>SO</b> — Sposta fuori	<b>DEL</b> — Cancella sostituendo
<b>SI</b> — Sposta dentro	

# TABELLA DELLE DIRAMAZIONI RELATIVE

## DIRAMAZIONE RELATIVA DIRETTA

LSD MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

## TABELLA DI DIRAMAZIONE RELATIVA INVERSA

LSD MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113
9	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97
A	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81
B	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
C	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
E	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
F	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

# LISTING DEL CODICE OPERATIVO ESADECIMALE

	LSD MSD	0	1	2	3	4	5	6	7
0		BRK	ORA-I, X				ORA-0, P	ASL-0, P	
1		BPL	ORA-I, Y				ORA-0, P, X	ASL-0, P, X	
2		JSR	AND-I, X			BIT-0, P	AND-0, P	ROL-0, P	
3		BMI	AND-I, Y				AND-0, P, X	ROL-0, P, X	
4		RTI	EOR-I, X				EOR-0, P	LSR-0, P	
5		BVC	EOR-I, Y				EOR-0, P, X	LSR-0, P, X	
6		RTS	ADC-I, X				ADC-0, P	ROR-0, P	
7		BVS	ADC-I, Y				ADC-0, P, X		
8			STA-I, X			STY-0, P	STA-0, P	STX-0, P	
9		BCC	STA-I, Y			STY-0, P, X	STA-0, P, X	STX-0, P, Y	
A	LDY-IMM	LDA-I, X		LDX-IMM		LDY-0, P	LDA-0, P	LDX-0, P	
B	BCS	LDA-I, Y				LDY-0, P, X	LDA-0, P, X	LDX-0, P, Y	
C	CPY-IMM	CMP-I, X				CPY-0, P	CMP-0, P	DEC-0, P	
D	BNE	CMP-I, Y					CMP-0, P, X	DEC-0, P, X	
E	CPX-IMM	SBC-I, X				CPX-0, P	SBC-0, P	INC-0, P	
F	BEG	SBC-I, Y					SBC-0, P, X	INC-0, P, X	

8	9	A	B	C	D	E	F	LSD MSD
PHP	ORA-IMM	ASL-A			ORA	ASL		0
CLC	ORA-Y	ROL-A			ORA-X	ASL-X		1
PLP	AND-IMM	ROL-A		BIT	AND	ROL		2
SEC	AND-Y				AND-X	ROL-X		3
PHA	EOR-IMM	LSR-A		JMP	EOR	LSR		4
CLI	EOR-Y				EOR-X	LSR-X		5
PLA	ADC-IMM	ROR-A		JMP-I	ADC	ROR		6
SEI	ADC-Y				ADC-X			7
DEY		TXA		STY	STA	STX		8
TYA	STA-Y	TXS			STA-X			9
TAY	LDA-IMM	TAX		LDY	LDA	LDX		A
CLV	LDA-Y	TSX		LDY-X	LDA-X	LDX-Y		B
INY	CMP-IMM	DEX		CPY	CMP	DEC		C
CLD	CMP-Y				CMP-X	DEC-X		D
INX	SBC-IMM	NOP		CPX	SBC	INC		E
SED	SBC-Y				SBC-X	INC-X		F

I = indiretto

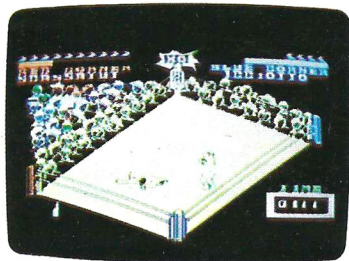
0 - P = pagina zero



# I SUPER-GAMES DI NEW VIDEO

## BOXE

MSX - Joystick in porta 1 o 2



Usa il joystick per muoverti intorno al ring e, tenendo premuto il tasto Fuoco, otterrai le varie mosse spostando la leva del joystick nelle otto direzioni.

Prima di iniziare il combattimento, dovrai scegliere il livello di difficoltà, il numero dei giocatori e la vitamina. Se selezionerai la vitamina su "on", sarai più forte durante il com-

battimento, ma nello stesso tempo più vulnerabile.

Il computer visualizzerà in alto la vitalità di ogni boxer, che diminuirà ad ogni colpo subito, fino a stabilire il Knock out del boxer. Se un round termina senza K.o., il computer darà un punteggio ai combattenti che stabilirà, alla fine dell'ultimo round, il vincitore.

## LAVANDERIA

MSX - Joystick in porta 1 o 2 - Tastiera

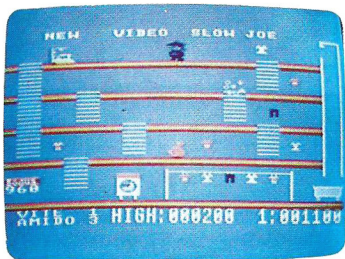
Tasti:

Tasti cursore: 4 direzioni

Spazio: lancia l'amido

Il lavandaio è nei guai! Non solo deve mandare avanti il lavoro in fretta, ma è cacciato da bolle di sapone velenose, ferri da stiro e assassini.

Aiuta il tuo amico a salvarsi da questa situa-



zione. Hai a disposizione una certa quantità di amido che, se lanciato contro gli inseguitori, li renderà immobili per poco tempo. Dovrai poi prendere un capo per volta e portarlo nella lavatrice. Usa l'amido con parsimonia, altrimenti avrai molti problemi...

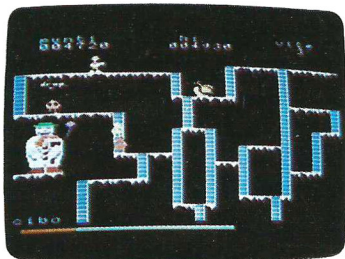
## PUPAZZO DI NEVE

**MSX - Joystick in porta 1 - Tastiera**

Tasti:

Tasti cursore: 4 direzioni

Spazio: Azione



Il gioco è composto dalle seguenti 4 fasi:

Fase 1:

raccogli 6 blocchi di neve in modo da erigere un pupazzo di neve. Porta ciascun blocco in corrispondenza della scritta "drop". Se incontrerai una fiammella mentre stai portando un blocco di neve, ti ritroverai con un po' di vapore da respirare! Se, invece, racco-

glierai un ghiacciolo, potrai congelare una fiammella ottenendo un bonus punti.

Fase 2:

raccogli la sciarpa, il cappello, il naso, i bottoni, gli occhi e il sorriso del pupazzo e componili su di esso. Cerca di evitare i mostri del sonno che ti faranno addormentare. Usa il ghiacciolo per salvarti da loro.

Fase 3:

Stesso gioco della fase 2, ma dovrai raccogliere una cravatta, gli occhiali, i pantaloni, la pila, lo skateboard e i palloncini.

Fase 4:

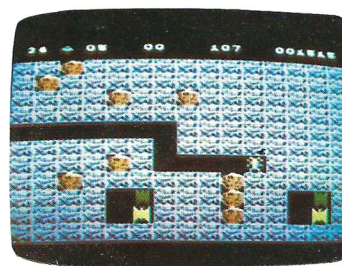
raccogli i 6 cubetti di ghiaccio per evitare che il pupazzo si sciolga.

## CAVERNE

**MSX - Joystick in porta 1 - Tastiera**

Tasti:

Tasti cursore: 4 direzioni



Dovrai disporre di tutto il tuo coraggio e abilità, mentre guidi il tuo eroe attraverso gli 84 differenti labirinti. Ti troverai faccia a faccia con pericolose farfalle, che dovrai trasformare in gioielli, fuochi mortali, e amebe dalla crescita incontrollata. Sarai in grado di capire come il muro incantato ti può dare i gioielli, oppure come poter entrare in quel box chiuso? Gioca e lo scoprirai!

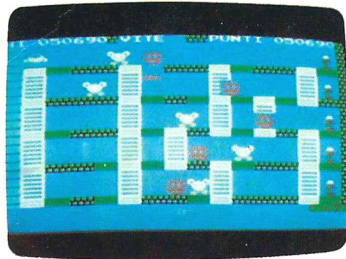
## AL FUOCO

**MSX - Joystick in porta 1 o 2 - Tastiera**

Tasti:

Tasti cursore - 4 direzioni e salto

Spazio - uso dell'estintore



Si salvi chi può dal palazzo in fiamme!!

Il tuo compito è quello di salvare i topolini dalle fiamme. Per poter far questo, dovrai avvicinarti al topo e condurlo verso il limite del piano in cui si trova per farlo cadere in prossimità dell'uscita. Prima di ogni partita, farai una prova di salvataggio senza le fiamme (pratica). Raccogli l'estintore che trovi ad ogni piano e usalo per spegnere le fiamme.

Ricordati che mentre porti un topo in salvo, non potrai usare né le scale né potrai saltare.

## CUBO

**MSX - Joystick in porta 1 o 2 - Tastiera**

Tasti:

Tasti cursore a coppia per ottenere una direzione diagonale

Muovi diagonalmente Jo verso l'alto o verso il basso atterrando su ogni cubo e cambian-



done il colore. Quando avrai ottenuto tutti i cubi dello stesso colore, passerai al livello successivo.

## OTELLO

**MSX - Joystick in porta 1 o 2 - Tastiera**

Tasti:

Cursore - 4 direzioni

Spazio - esecuzione



Dovrai spremere al massimo le tue abilità strategiche ed intellettive in un gioco senza pari.

Lo scopo del gioco è quello di riempire il casellario col maggior numero di pedine.

Si fa una mossa alla volta ed è obbligatorio mangiare almeno una pedina dell'avversario, orizzontalmente, verticalmente, diagonalmente.

Buona fortuna!

IN TUTTE LE EDICOLE D'ITALIA



Estasy Rosa

I NUOVI ROMANZI D'AMORE  
CHE RACCONTANO **TUTTO**

3370144



Estasy Rosa