# ZX MASTER MIND


**Master Mind. Il gioco in scatola**
Il gioco, nella versione classica, consiste di una tavola di gioco e da pioli colorati di due tipi:
- Il primo tipo, sono quelli più grandi e sono di 6 colori diversi, vengono usati per la composizione del codice segreto e dei codici tentativi.
- Il secondo tipo, sono più piccoli e di colore bianco e nero, vengono usati per la composizione dei codici chiave.
(Nell'immagine sopra la mia scatola nella variante 8 colori comprata nei primi anni 80')

**Regolamento del gioco**
Il Master Mind è un gioco di logica dove due giocatori si alternano nei ruoli di Codificatore e Decodificatore.
Il Codificatore è colui che sceglie e nasconde il codice segreto e il Decodificatore è il giocatore che lo deve indovinare.
L'obbiettivo è quello di indovinare il Codice Segreto nel minore numero di tentativi possibili.
I due giocatori decideranno all'inizio quante partite effettuare e al termine confronteranno la somma dei tentativi. Chi avrà il numero minore, vince.
Il codice segreto è composto da quattro pioli colorati tenuti nascosti da un coperchietto della tavola di gioco.
Il Decodificatore per tentativi successivi giocherà dei codici, anch'essi composti da pioli colorati e il Codificatore porrà dei Pioli Chiave di colore Bianco e Nero per ogni tentativo.
In base alle risposte del Codificatore, il Decodificatore deciderà il Codice successivo da giocare.
I Codici Chiave in risposta ad ogni tentativo seguono queste regole:
- Tanti pioli neri quanti sono i colori del Codice tentativo che sono presenti nel Codice Segreto e che sono anche nella posizione giusta.
- Tanti pioli bianchi quanti sono i colori presenti nel Codice Segreto ma che sono nella posizione sbagliata.
- Nessun Piolo Chiave per i colori del codice giocato non presenti nel Codice Segreto
- Nota: I Pioli Bianchi e Neri indicano la quantità di pioli colorati presenti nel Codice Segreto ma non la loro posizione.

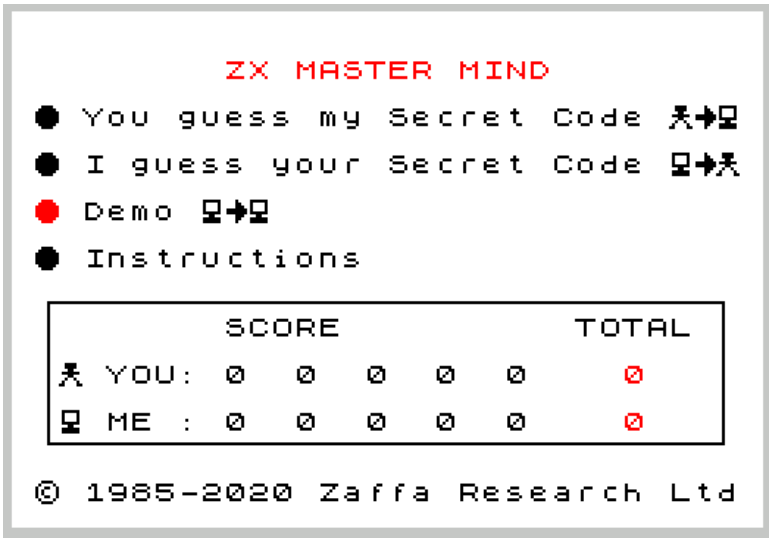
Per evitare errori nelle risposte in caso di colori ripetuti, è preferibile che il Codificatore, metta prima i pioli neri e poi quelli bianchi facendo attenzione per quest'ultimi di escludere i colori già considerati nella risposta dei pioli neri.
Nel seguente esempio il Codificatore mette un solo piolo nero per il blu presente e in posizione giusta. Nessun piolo bianco per l'altro blu.
Nessun altro piolo per gli altri colori in quanto non sono presenti nel codice segreto

**Possibili combinazioni di codici segreti.**
Nella versione classica, il numero totale dei possibili Codici Segreti è calcolato secondo la formula del calcolo combinatorio delle Disposizioni con Ripetizione: $D=n^k$ dove nel nostro caso, n è il numero di colori e k il numero di posizioni dove i colori possono essere collocati, quindi $6^4$ e cioè 1296 possibilità di codici segreti.
Per comodità, anche se impropriamente, tali disposizioni verranno chiamate combinazioni.
Esistono altre varianti del gioco con più colori e posizioni o versioni facilitate con meno colori/posizioni per bambini.
Nella versione classica i colori sono bianco, blu, giallo, nero, verde. La variante a 8 colori aggiunge l'arancione e il marrone.
Rispetto ai colori del gioco in scatola, i colori usati dal programma, sono quelli dello ZX Spectrum e cioè: blu, rosso, magenta, verde, azzurro e giallo. I colori bianco e nero vengono usati solo per i codici chiave.

**Modalità di gioco e opzioni del programma "ZX Master Mind"**
Dal menu del programma sono disponibili le seguenti opzioni:

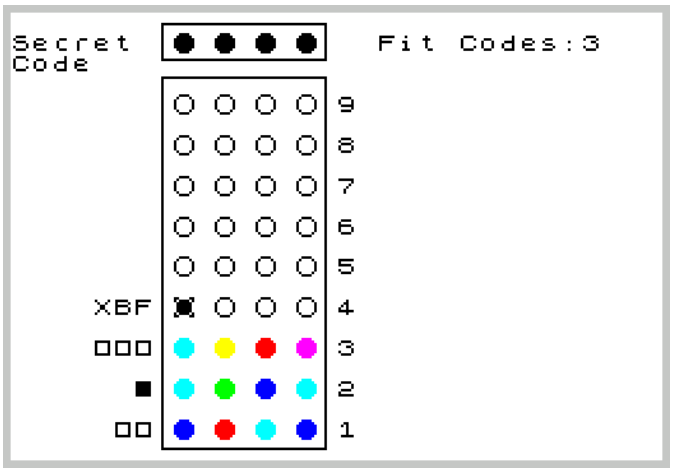
- Il giocatore indovina il codice segreto scelto in modo casuale dal programma
- Il programma indovina il codice segreto scelto dal giocatore
- Modalità Demo – Il programma indovina un codice segreto casuale
- Istruzioni

Sotto al menu è visualizzato il punteggio per le partite giocate tra il giocatore e il programma.



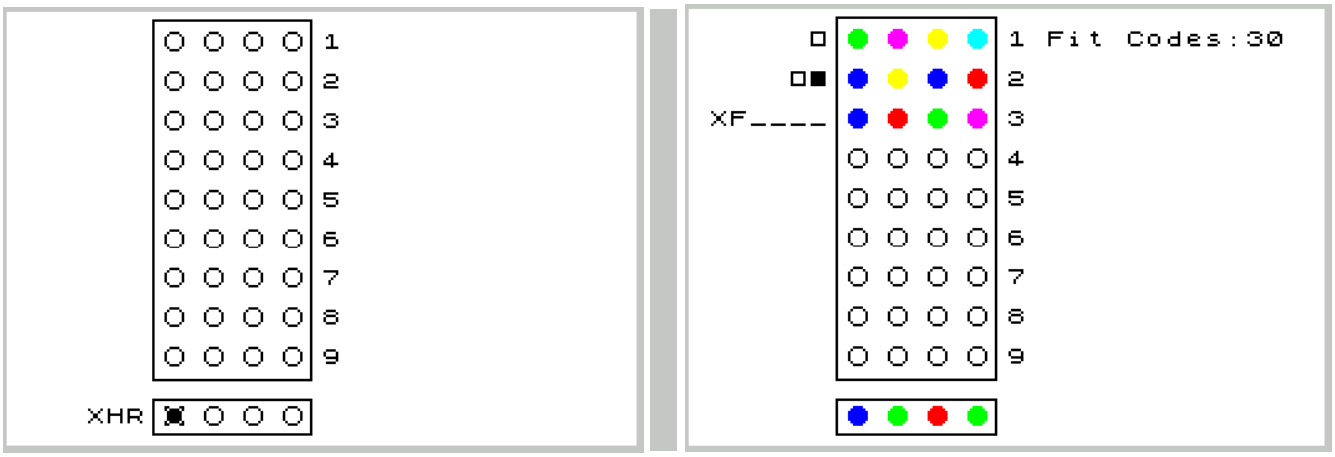
**Di seguito i dettagli per ogni opzione:**

<u>Il giocatore indovina il codice segreto</u>**.**
Il giocatore è il decodificatore e il programma il codificatore
Il programma sceglierà un codice segreto casuale che terrà oscurato nella parte alta della tavola.
Ad ogni codice tentativo del giocatore, il programma attribuirà un codice chiave (pioli bianchi e neri).
L'opzione "X" abbandona la partita e si ritorna al menu.
Per le opzioni "B" ed "F", vedere il paragrafo: "Opzioni F e B"
In alto a destra vengono visualizzati i "Fit Codes" che sono il numero dei codici segreti rimanenti dopo ogni tentativo.



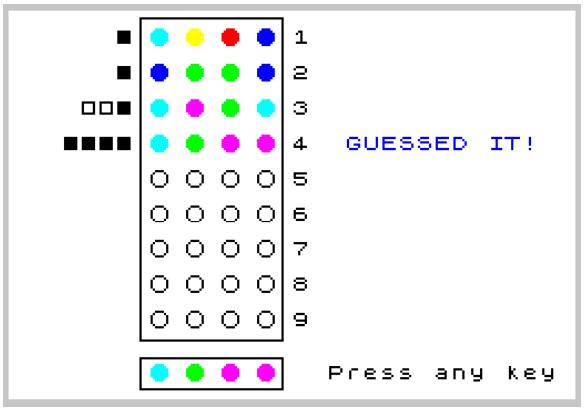

<u>Il programma indovina il codice segreto.</u>
Il programma è il decodificatore e il giocatore il codificatore.
All'inizio della partita il giocatore ha tre possibilità per sceglie il codice segreto:

1) Scegliere un colore per ogni posizione
2) Usare l'opzione "R" (Random) per generare un codice segreto casuale
3) Usare l'opzione "H" (Hidden) per non inserire il proprio codice segreto ma scriverlo su un foglio.

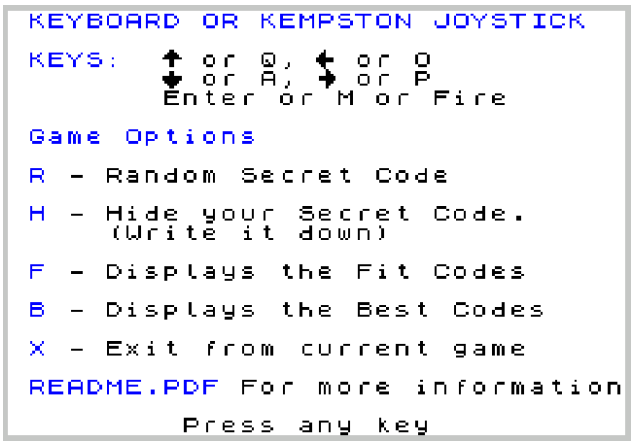



Dopo aver inserito il codice segreto, il programma inizierà a giocare il primo tentativo a cui si dovrà fornire un codice chiave.
L'opzione "X" abbandona la partita.
L'opzione "F" verrà trattata nel paragrafo: "Opzioni F e B"

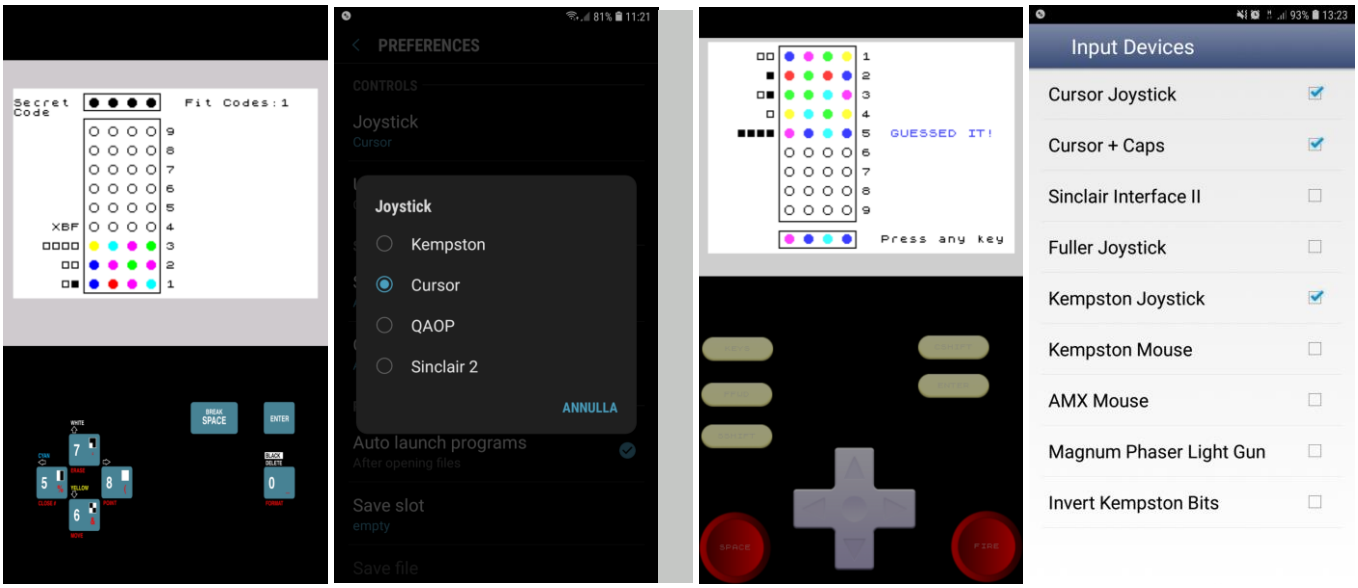
**Il programma indovina un codice segreto casuale – Modalità Demo**
Il programma giocherà sia da Codificatore che da Decodificatore. Sceglierà in modo casuale un Codice Segreto e giocherà fino a quando non lo indovina. Questa modalità può essere usata per capire come si svolge il gioco.




**Istruzioni. Uso della tastiera e del Joystick Kempston**
Per selezionare i pioli colorati e le opzioni del programma, usare i tasti cursori o il Joystick Kempston.
La scelta viene confermata con il tasto “Invio” o “Fire” del Joystick.
Giocando con un Spectrum con tasti in gomma è meglio usare i tasti alternativi: Q A O P come tasti cursore e M come INVIO.



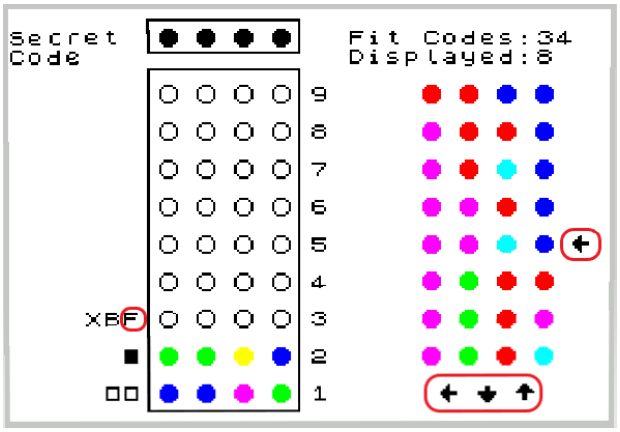

Utilizzando un emulatore dello ZX Spectrum per Android è preferibile scegliere come dispositivo di input nelle preferenze i "tasti cursore". L’immagine che segue fa riferimento a: “Unreal Speccy Portable” e a “Speccy”



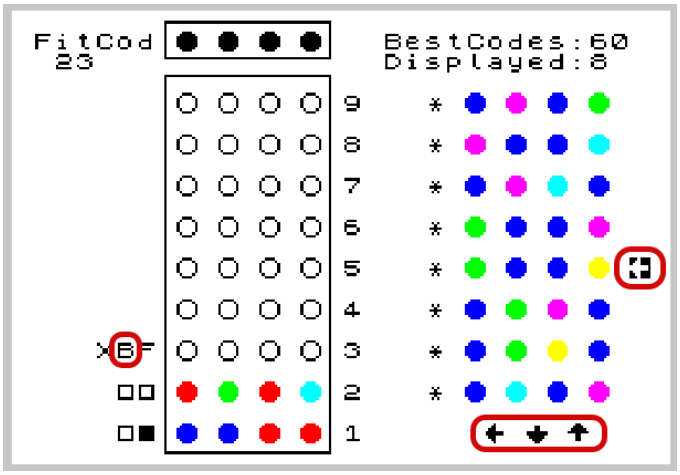
**Caricare ed eseguire il programma**.
Per gli emulatori ZX Spectrum, aprire il file auto avviante MMEMUXX.TAP (XX è la versione).
È stato testato con Fuse per Windows, Speccy e Unreal Speccy per Android.
Per lo ZX Spectrum, il classico LOAD "" e poi riprodurre il file MMVAWXX.WAV da un lettore MP3/Smartphone/PC collegato tramite cavo Jack alla porta "EAR" dello Spectrum. Testato sul mio ZX Spectrum +.

**Opzioni “F” e “B”**
Queste opzioni non sono necessarie per il normale svolgimento del gioco, tuttavia posso essere usate se si è in difficoltà a trovare un codice tentativo oppure per migliorare la propria strategia di gioco.

Opzione “F” (Fit Codes)
Questa opzione visualizza i codici segreti possibili chiamati dal programma Fit Codes. Tra questi c’è il reale codice segreto.
In alto a destra è visualizzato il totale dei Fit Codes e quanti sono stati visualizzati.
Sono visualizzati a gruppi da otto. Si passa al gruppo successivo usando il tasto freccia “Giù”.
La freccia “Su” abilita un cursore che può scorrere la lista dei fit codes per scegliere il codice tentativo che deve poi essere confermato con il tasto INVIO.
Per tornare indietro, premere il tasto cursore con freccia a sinistra.




Opzione “B” (Best Codes)
Visualizzazione dei Best Codes. I Best Codes sono selezionati da un algoritmo che individua i codici migliori da giocare.
Durante l’elaborazione, un contatore indica la progressione dei codici da processare.
Giocare i Best Codes, in termini probabilistici, riduce il numero di tentativi per indovinare il codice segreto.
Se i Best Codes sono contrassegnati con un asterisco, sono di tipo inconsistente, cioè non potranno indovinare il codice segreto nel momento in cui vengono giocati ma offrono comunque migliori probabilità di arrivare al codice segreto con un minor numero di tentativi rispetto alla scelta di un codice tentativo tra fit codes.
In alto a destra è visualizzato il totale dei Best Codes e quanti di questi sono stati visualizzati.
Per navigare tra Best Codes e fare la selezione, le modalità sono le stesse come per i Fit Codes



**Come funziona l'algoritmo "Best Codes"**

Iniziamo a capire come vengono individuati i Fit Codes assumendo che il programma che gioca come decodificatore.
1) All'inizio il programma genera tutti i 1296 codici che tiene in memoria. In questa fase questi codici sono tutti Fit Codes.
2) Alla prima giocata il programma seleziona un Fit Code in modo casuale ad esclusione per i codici con un colore ripetuto 3 o 4 volte. Sono i codici peggiori per iniziare.
3) Confronta il suo Codice tentativo con ognuno dei Fit Codes. Tutti quelli che generano un Codice chiave diverso da quello ottenuto dal giocatore, vengono scartati.
4) Il programma gioca un Fit Code di quelli rimasti.
5) Il programma ripete il procedimento dal punto 3).

Ora, premesso che ogni Fit Code ha la stessa probabilità degli altri Fit Codes di indovinare il Codice Segreto, nel caso però che non lo indovini, in base al Codice Chiave ricevuto rimarranno tanti o meno Fit Codes per il tentativo successivo.
L'obbiettivo dell'algoritmo è quello di far rimanere, in termini probabilistici, il minor numero di Fit Codes.
Ovviamente, la componente fortuna non può essere eliminata e mantiene un ruolo significativo.

Prima versione dell'algoritmo**:**

Sviluppai questa versione nel 1985. È la routine "Simula" inclusa nei vecchi listati che riporto nelle immagini più avanti. Nel programma attuale questa routine è stata migliorata.
Una copia della lista dei Fit Codes viene usata come lista dei Codici Segreti possibili.
Per ogni Fit Code viene associato un punteggio che viene determinato come segue: viene simulato di giocare il primo Fit Code e viene ipotizzato che il Codice Segreto sia il primo della sua lista. In base al Codice Chiave generato vengono contati tutti i Fit Codes compatibili con il Codice Chiave. Il numero ottenuto va a sommarsi al punteggio del primo Fit Code.
Questa operazione viene ripetuta ipotizzando che il codice segreto sia il secondo dalla lista ottenendo un nuovo numero da sommare al punteggio del primo Fit Code.
Terminati tutti i Codici Segreti, si passa al secondo Fit Code e si ripete la simulazione per tutti i Codici Segreti per ottenere il punteggio anche per il secondo Fit Code.
Il procedimento sopra, viene ripetuto per i Fit Codes rimanenti.
Al termine, i Fit Codes con il punteggio più basso saranno eletti Best Codes.
Se ci saranno più Fit Codes con lo stesso punteggio, saranno tutti eletti a Best Codes.
Questo algoritmo indovina il Codice Segreto in media alla 4,54 giocata. Risultato ricavato da 1000 partite giocate.

Versione migliorata:

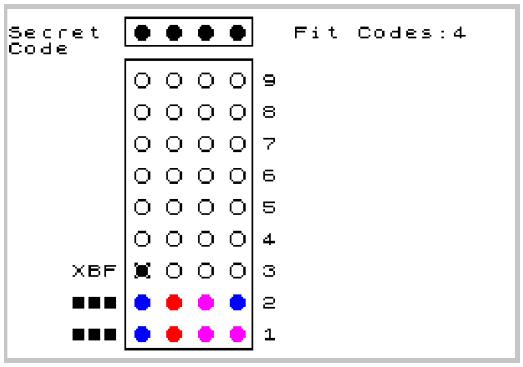
Viene fatta una copia della lista dei Fit Codes e questa viene considerata la lista dei Codici Segreti Possibili.
Ora viene simulato di giocare il primo Fit Code della lista e che il Codice Segreto sia il primo della rispettiva lista. In questo caso, poiché i due codici sono identici, il Codice Chiave sarà 4 neri.
Per ogni Fit Code, viene preventivamente definita una tabella come quella in figura sotto che contiene tutti i Codici Chiave possibili e ogni casella è valorizzata a zero.
In questo caso verrà incrementata casella "4 neri" di uno.

| | 0 Neri | 1 Nero | 2 Neri | 3 Neri | 4 neri |
|---|---|---|---|---|---|
| 0 Bianchi | 0 | 0 | 0 | 0 | 0 |
| 1 Bianco | 0 | 0 | 0 | | |
| 2 Bianchi | 0 | 0 | 0 | | |
| 3 Bianchi | 0 | 0 | | | |
| 4 Bianchi | 0 | | | | |

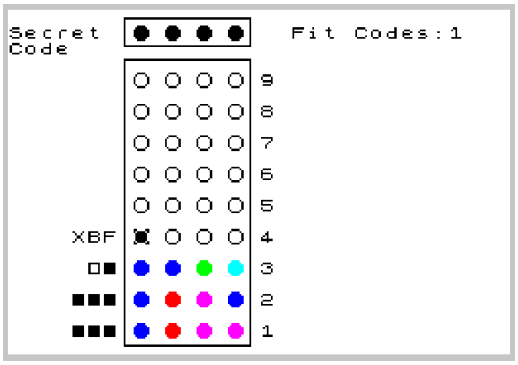
L'operazione viene ripetuta tenendo fisso il primo Fit Code e passando al secondo Codice Segreto.
Anche qua, in base al Codice Chiave ottenuto, verrà incrementa la casella corrispondente.
Il processo viene ripetuto per tutti i Codici Segreti.
Al termine, i valori nelle caselle indicheranno il numero dei Fit Codes che rimarrebbero se venisse giocato il primo Fit Code e il Codice Chiave fosse quello relativo alla casella stessa.
Ad esempio, se la casella 2 Neri e 2 Bianchi conterrà 3, significa che giocando il primo Fit Code e ricevendo come Codice Chiave 2 Neri e 2 Bianchi, i Fit Codes rimanenti per la giocata successiva saranno 3. (Ovviamente se non viene indovinato il codice segreto)

A questo punto, l'algoritmo associa al primo Fit Code il valore massimo (MAX) che trova in tabella.
MAX è quindi il numero massimo di Fit Codes che possono rimanere se viene giocato il primo Fit Code.
L'algoritmo ripete ora lo stesso processo per il Fit Code successivo, ottenendone il relativo MAX.
Continua così fino al termine di tutti i Fit Codes.
Ora verranno controllati tutti i Fit Codes. Quello che avrà il valore MAX più basso, sarà eletto Best Code.
Se altri Fit Codes avranno lo stesso MAX, anche questi saranno Best Codes.
La base di elaborazione dei Fit Codes può essere estesa a tutti i codici (1296) mettendo in concorrenza i Fit Codes con i codici inconsistenti, (codici che non possono essere il codice segreto). Ebbene, talvolta i codici inconsistenti hanno un MAX minore di quello ottenuto dai Fit Codes, e quindi potranno essere giocati lasciando un minor numero di Fit Codes per il tentativo successivo.
Se parte dei Fit Codes e alcuni codici inconsistenti riceveranno lo stesso valore di MAX, solo i primi verranno eletti a Best Codes quanto questi hanno anche la possibilità di indovinare il codice segreto.
Poiché il tempo di elaborazione aumenta in base al numero di codici processati, occorre limitare l'uso dell'algoritmo per evitare tempi di attesa eccessivi che possano annoiare il giocatore.
In caso il programma è decodificatore, l'algoritmo viene eseguito se i fit codes sono inferiori o uguali a 317.
Se i fit codes sono minori o uguali a 50, l'algoritmo considera tutti i 1296 codici per l'elaborazione.
Stesse modalità sopra quando è il giocatore ad essere il decodificatore e seleziona l'opzione "B".
Scegliendo come primo tentativo un codice che non sia un colore ripetuto 4 volte, rimangono al massimo 317 Fit Codes e quindi, a meno del caso sopra, al secondo tentativo può essere eseguito l'algoritmo.
Quando poi nei tentativi successivi il numero di Fit Codes scenderà, 50 sarà un buon compromesso per estendere l'uso dell'algoritmo a tutti i 1296 codici con la possibilità di avere come Best Codes anche codici inconsistenti.

**Esempio di utilizzo di un codice inconsistente.**
Ipotizza di esser giunto a questa situazione di gioco:

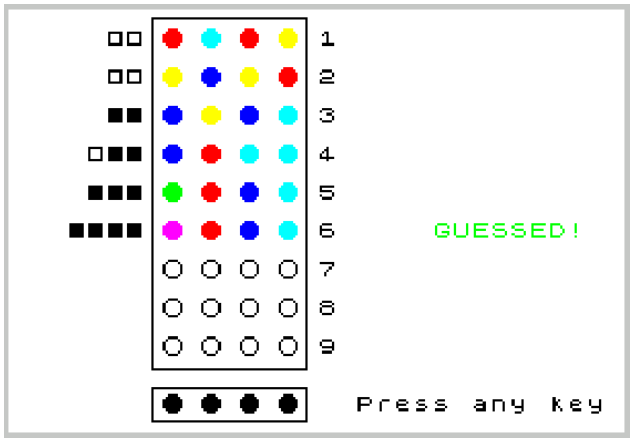



Non sembra male, 3 neri al primo tentativo e anche al secondo che confermano che i primi 3 colori sono giusti e anche al posto giusto. A questo punto basta ripetere il codice cambiando l'ultimo colore con uno dei 4 possibili. Così facendo però, hai il 25% di probabilità di indovinare il codice segreto al terzo tentativo e se sei sfortunato, lo indovinerai al sesto tentativo.
Se invece giochi un codice inconsistente come quello nella figura sotto:



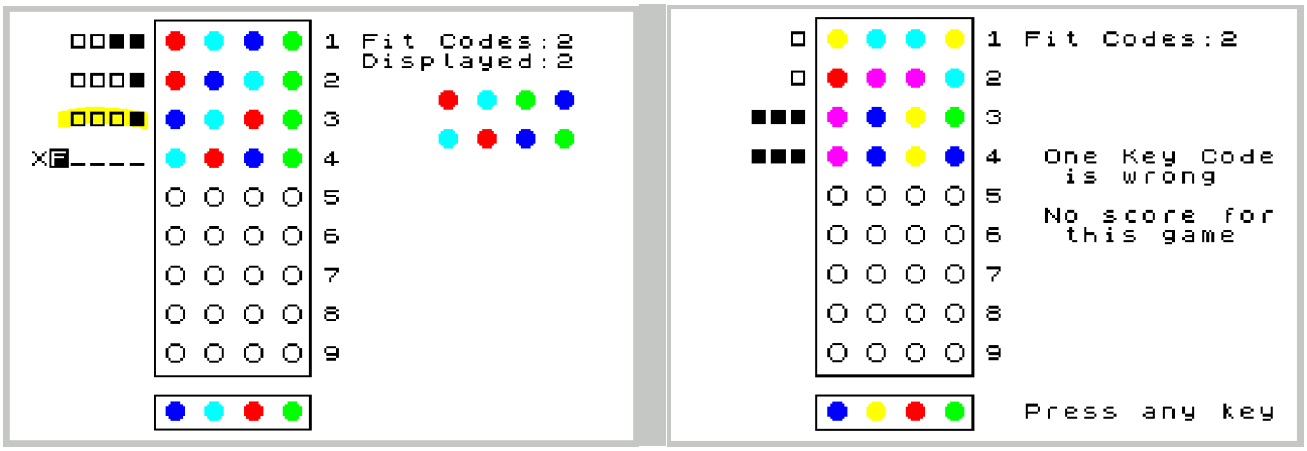

Avrai la certezza di indovinare il codice segreto al quarto tentativo.
Al quinto tentativo, se nella medesima situazione il primo codice giocato sono 4 colori diversi.

**Quanti tentativi servono al programma per indovinare il codice segreto?**
Al massimo 5. La media è di 4,47. Il programma potrebbe far meglio ma i tempi di attesa per l'elaborazione dei codici si allungherebbe come spiegato in precedenza. In rari casi e solo se il tentativo iniziale è del tipo 1123, cioè con un solo colore ripetuto, potrebbero essere necessari 6 tentativi come nell'esempio sotto.




I tentativi iniziali di tipo 1112 e 1111, sono quelli che implicano un maggior rischio di arrivare ad indovinare il codice segreto al sesto tentativo o oltre e quindi non sono mai giocati dal programma.

**Si può barare dando codici chiave errati?**
Nell'immagine sotto, il programma indovina il codice segreto al terzo tentativo ma il giocatore bara rispondendo con tre bianchi e un nero. Poiché questo codice chiave lascia ancora due possibilità, il programma procede con il tentativo successivo.
Nel caso invece, un codice chiave errato non lascia tentativi possibili, il gioco si interrompe e un messaggio avvertirà il giocatore che ha sbagliato almeno un Codice Chiave (Key Code).



**“ZX Master Mind”. Breve storia di come è andata.**

Negli anni 80’, microcomputer come ZX Spectrum e Commodore 64, diedero alla mia generazione la possibilità di avere un computer a casa senza spendere grandi cifre e poter imparare a programmare, anche se la grande diffusione di questo tipo di computer era più legata alla possibilità di utilizzarsi come console per giochi. Giochi che si potevano acquistare in originale o trovare in audio cassette allegate a riviste acquistate in edicola.

Iniziai a scrivere il programma per il Master Mind nel 1985 quando ero studente alle scuole superiori a indirizzo Informatico. Trovavo sfidante implementare nei programmi la logica dei giochi come ad esempio Battaglia Navale, Forza 4 e Tetris. Scrivevo la parte della logica dei giochi, quella che mi dava più soddisfazione, ma poi perdevo interesse e i giochi rimanevano incompleti.

Master Mind fu quello che mi impegnò per più tempo perché l’obbiettivo di fare un programma che potesse giocare meglio di un buon giocatore umano, mi piaceva molto.

Da ciò avrei cercato di imparare le strategie di gioco migliori per diventare più bravo anzi il più bravo tra gli amici. :-).

Non sapevo fin dove sarei riuscito ad arrivare, ma mi accorsi ben presto che non sarebbe stato facile perseguire il mio obbiettivo. Il codice sviluppato in Basic era troppo lento per le elaborazioni che avevo in mente e così la parte di calcolo dovetti convertirla in Assembler.

Un'altra sfida era la gestione della memoria, dovevo riuscire a far stare in memoria il programma in basic, il sorgente in assembler e il compilato, due assemblatori: Zeus, che ho usato per scrivere l'assemblatore e Champ, per la fase di debug istruzioni per istruzione e infine le 1296 combinazioni che occupavano circa 13K.

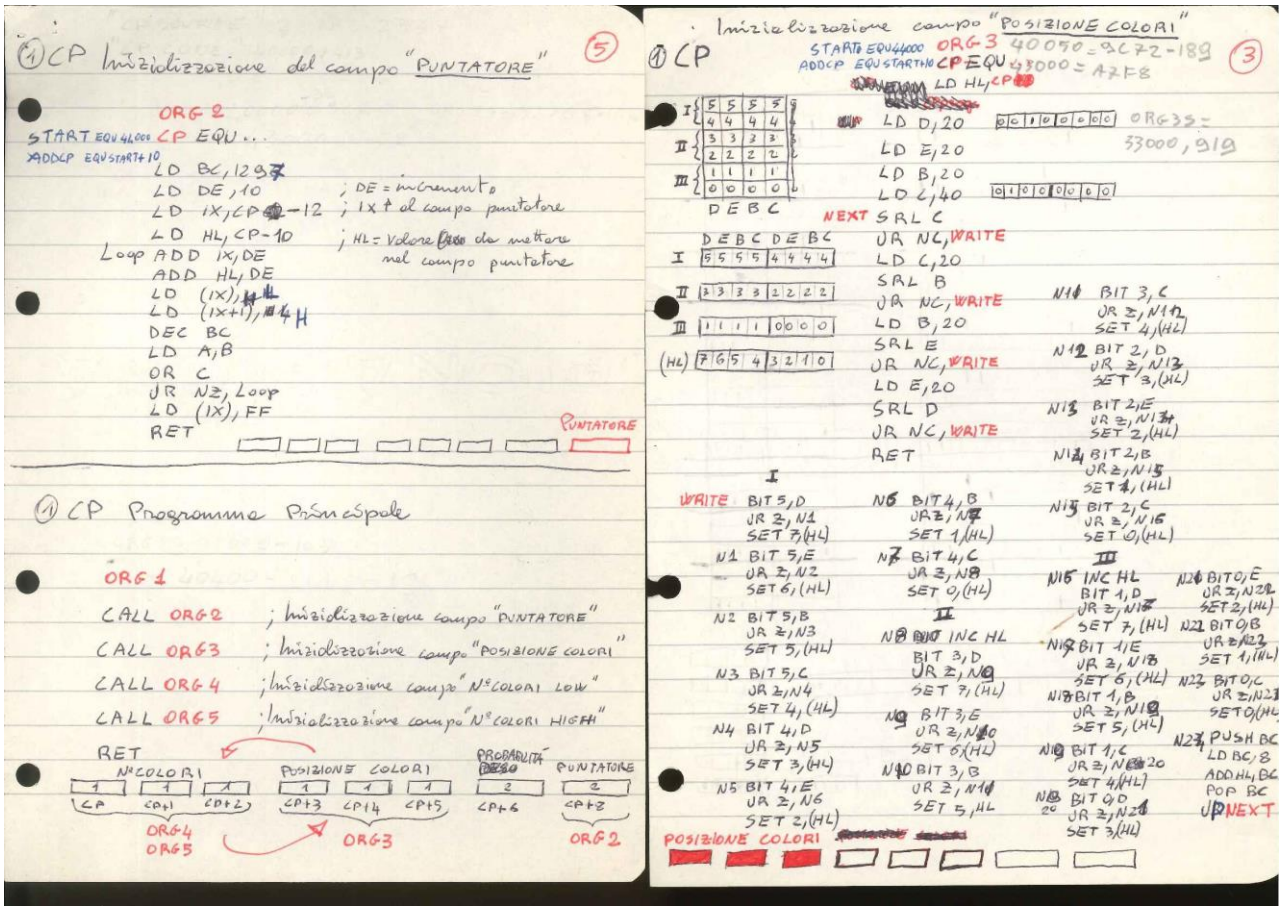
Alla fine, il “cuore” del programma, che erano poi le routine in Assembler c’erano. Ma come per altri programmi, rimase incompleto.

Da allora non ho più fatto nulla e le uniche volte che lo Spectrum mi è capitato per le mani, è stato per spostarlo da cantine a soffitte.

"Prima o poi mi piacerebbe accenderlo e farci qualcosa", era il pensiero di quando lo vedevo. Poi finalmente un po' di tempo fa è arrivato il momento giusto mentre facevo pulizia tra vecchie scartoffie, ritrovai i vecchi listati di programmi tra cui quelli delle routine in assembler del Master Mind. Speravo di ripartire dal programma che avevo lasciato, ma accendendo il mio Spectrum mi accorsi subito che aveva problemi e quindi dovetti rinunciare.

Decisi allora di trovare un emulatore dello Speccy su PC, per riscrivere il programma partendo dalla base dei listati ritrovati.

Ecco i listati delle routine. C’è anche la struttura dati dei codici e come organizzavo la memoria per contenere il codice sorgente, il compilato e i due Assemblatori:



## TEST2 Numero di X+0

| ENTRATE | USCITE | MODIFICA: |
|---|---|---|
| IX ↑ CP<br>IY ↑ C<br>D = F0<br>E = 0F | C = 0+X<br>A = 0+X | AF, BC, DE |

```
ORG 7
;LD C,0 ;
;LD D,F0 ;Mascheramento
;LD E,0F ;NIBBLE non interessato
LD A,D ;I Nibble
AND (IX+3)
LD B,A
LD A,D
AND (IY+3)
CP B
JR C,l1
LD A,B
l1 SRLA SRLA SRLA SRLA
LD C,A
LD A,E ;II Nibble
AND (IX+3)
LD B,A
LD A,E
AND (IY+3)
CP B
JR C,l2 Salto se A<B
LD A,B
l2 ADD C
LD C,A
LD A,D ;III Nibble
AND (IX+4)
LD B,A
LD A,D
AND (IY+4)
CP B
JR C,l3
LD A,B
l3 SLA SLA SLA SLA
ADD C
LD C,A
LD A,E ;IV Nibble
AND (IX+4)
LD B,A
LD A,E
AND (IY+4)
CP B
JR C,l4
LD A,B
l4 ADD C
LD C,A
LD A,D ;V Nibble
AND (IX+5)
LD B,A
LD A,D
AND (IY+5)
CP B
JR C,l5
LD A,B
l5 SLA A x4
l6 ADD C
LD C,A
LD A,E ;VI Nibble
AND (IX+5)
LD B,A
LD A,E
AND (IY+5)
CP B
JR C,l6
LD A,B
l6 ADD C
LD C,A
RET
```

C: IY+3 (I II), IY+4 (III IV), IY+5 (V VI)

CP: IX+3 (I II), IX+4 (III IV), IX+5 (V VI)

Nº COLORI

## CPR Eliminazione CP impossibili

| ENTRATE | USCITE | MODIFICA: |
|---|---|---|
| (C),<br>(X) e (0+X) | / | tutti i Reg<br>e i Buffer ↑ |

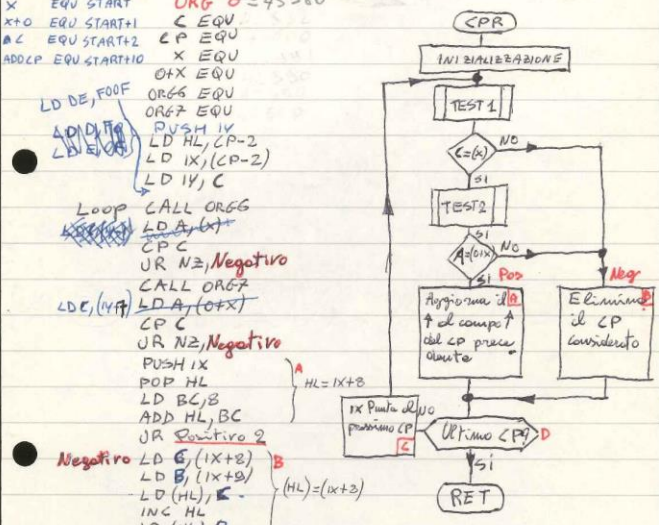
```
START EQU 46000
X EQU START
X+0 EQU START+1
C EQU START+2
ADDCP EQU START+10
ORG 8 = 43580
C EQU
CP EQU
X EQU
0+X EQU
ORG6 EQU
ORG7 EQU
PUSH IY
LD HL,CP-2
LD IX,(CP-2)
LD IY,C
LD DE,FOOF
Loop CALL ORG6
LD A,(X)+
CP C
JR NZ,Negativo
CALL ORG7
LD E,(IY+)
LD A,(0+X)
CP C
JR NZ,Negativo
PUSH IX
POP HL
LD BC,8
ADD HL,BC
JR Positivo 2
Negativo LD C,(IX+8)
LD B,(IX+9)
LD (HL),C
INC HL
LD (HL),B
DEC HL
Positivo 2 LD A,FF
XOR (IX+8)
JR NZ,NORET
RET Z
POP IY
NORET LD C,(IX+8)
LD B,(IX+9)
PUSH BC
POP IX
JR Loop
```

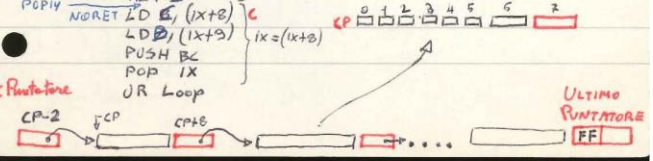
A: HL = IX+8

B: (HL) = (IX+8)

C: IX = (IX+8)

D: Ritorna se i CP sono finiti




I Puntatore




ULTIMO PUNTATORE

## (1) CP Inizializzazione dei campi "Nº colori"

ORG 4 = 34000 − 239

(4)

```
START EQU 44000
ADD CP EQU START+10
```

CP, CP+1, CP+2

### NIBBLES LOW

```
ORG 4
CP EQU ...
LD HL,CP+3 ;ANº colori
LD IX,CP ;APos colori
Loop LD C,3
Repeat LD B,4
XOR A
LD D,(IX)
Nextbit SRL D
JR NC,Noset
INC A
Noset DJNZ Nextbit
LD (HL),A
INC HL
INC IX
DEC C
JR NZ,Repeat
LD A,FF
XOR (IX+5)
RET Z
PUSH BC
LD BC,7
ADD HL,BC
ADD IX,BC
POP BC
JR Loop
```

### NIBBLES HIGH

```
ORG 5 40300 − 55 9D6C
CP EQU 43000
LD HL,CP+3
LD IX,CP
ORG 55 = 34 240 − 275
Loop LD C,3
Repeat LD B,4
XOR A
LD D,(IX)
Nextbit SLA D
JR NC,Noset
INC A
Noset DJNZ Nextbit
LD E,(HL)
SLA A
SLA A
SLA A
SLA A
ADD A,E
LD (HL),A
INC HL
INC IX
DEC C
JR NZ,Repeat
LD A,FF
XOR (IX+5)
RET Z
PUSH BC
LD BC,7
ADD HL,BC
ADD IX,BC
POP BC
JR Loop
```

DEC B ?

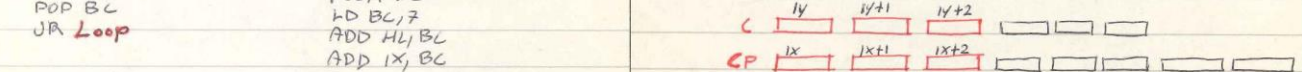
Nº COLORI

Nº COLORI

## TEST1 Numero di X

| ENTRATA | USCITE | MODIFICA: |
|---|---|---|
| IX ↑ CP<br>IY ↑ C | C = NºX | AF, BC |

```
ORG 6
LD BC,0 ; C = contatore delle X
LD A,(IX)
AND (IY)
LD B,A
LD A,(IX+1)
AND (IY+1)
OR B
LD B,A
LD A,(IX+2)
AND (IY+2)
OR B
LD B,8
Next SLA A ; Gli 1 in A sono le X
JR NC,l1
INC C
l1 DJNZ,Next
RET
```




POSIZIONE COLORI

## SIMULA

Calcola il peso di ogni CP | ENTRATE / | USCITE / | MODIFICA tutti i registri e i campi "PESO"

1

```
ORG9
X     EQU
CP    EQU
O+X   EQU
ORG6  EQU
ORG7  EQU
      LD IY,CP
NEXT3 LD IX,(CP-2)
Next2 CALL ORG6
      LD A,C        ;L=N°X
      LD (X),A
      CALL ORG7
      LD A,C        ;H=N°O+X
      LD (O+X),A
      PUSH IX
      LD IX,(CP-2)
Next1 CALL ORG6
      LD A,(X)
      CP C
      JR NZ,Negativo
      CP 4
      JR Z,Negativo
      CALL ORG7
      LD A,(O+X)
      CP C
      JR NZ,Negativo
      LD B,(IX+6)
      LD C,(IX+7)
      INC BC
      LD (IX+6),B
      LD (IX+7),C
Negativo LD A,FF
      XOR (IX+8)
      JR Z,Exit1
```

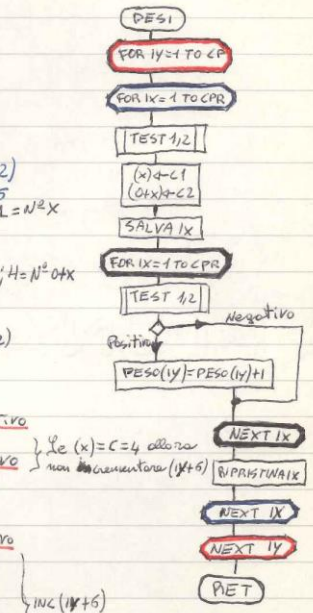
LD DE,FOOF
PUSH IY
LD H,A
LD L,C
CPH
LA A,L

Se (x)=C=4 allora non incrementare (IX+6)

INC (IX+6)

Se (IX+8)=FF Esci

CONTINUA




## SIMULA (continuazione)

2

```
      LD B,(IX+8)
      LD C,(IX+8)
      PUSH BC
      POP IX
      JR Next1
Exit1 POP IX
      LD A,FF
      XOR (IX+8)
      JR Z,Exit2
      LD B,(IX+9)
      LD C,(IX+8)
      PUSH BC
      POP IX
      JR Next2
Exit2 LD A,FF
      XOR (IY+8)
      JR Z,FINE
      LD BC,10
      ADD IY,BC
      JR Next3
FINE  RET
```

IX=(IX+8)

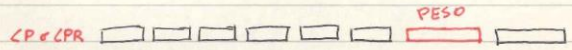
Se (IX+8)=FF Esci

IX=(IX+8)

Se (IY+8)=FF Finisci

ELSE IY=IY+10 e Ripeti

POP IY




## MASTER MIND — STRUTTURA DEL FILE

START = 44000 ABE0

| | | | | |
|---|---|---|---|---|
| START | I NIBBLE | II NIBBLE | 0 E0 | POSIZIONE COLORI |
| START+1 | III NIBBLE | IV NIBBLE | 1 E1 | |
| START+2 | V NIBBLE | VI NIBBLE | 2 E2 | |
| START+3 | I NIBBLE | II NIBBLE | 3 E3 | NUMERO COLORI |
| START+4 | III NIBBLE | IV NIBBLE | 4 E4 | |
| START+5 | V NIBBLE | VI NIBBLE | 5 E5 | |
| START+6 | BYTE | | 6 E6 | X |
| START+7 | BYTE | | 7 E7 | O+X |
| START+8 | ADDRESS | | 8 E8 | PUNTATORE 0 |
| START+9 | | | 9 E9 | |
| START+10 | 0 I NIBBLE | II NIBBLE | 10 EA | POSIZIONE COLORI |
| START+11 | 1 III NIBBLE | IV NIBBLE | 11 EB | |
| START+12 | 2 V NIBBLE | VI NIBBLE | 12 EC | |
| START+13 | 3 I NIBBLE | II NIBBLE | 13 ED | NUMERO COLORI |
| START+14 | 4 III NIBBLE | IV NIBBLE | 14 EE | |
| START+15 | 5 V NIBBLE | VI NIBBLE | 15 EF | |
| START+16 | 6 WORD | | 16 F0 | PESO |
| START+17 | 7 | | 17 F1 | |
| START+18 | 8 ADDRESS | | 18 F2 | PUNTATORE 1 |
| START+19 | 9 | | 19 F3 | |

C = VARIABILI

CP = RECORD

PUNTATORE INIZIO FILE

### VARIABILI

| 0-E0 | 1-E1 | 2-E2 | 3-E3 | 4-E4 | 5-E5 | 6-E6 | 7-E7 | 8-E8 9-E9 |
|---|---|---|---|---|---|---|---|---|
| I II III IV V VI | I II III IV V VI | | | | | X | O+X | PUNTATORE 0 |

### PUNTATORE

### RECORD

| 10-EA | 11-EB | 12-EC | 13-ED | 14-EE | 15-EF | 16-F0 | 17-F1 | 18-F2 19-F3 |
|---|---|---|---|---|---|---|---|---|
| CP I II III IV V VI | I II III IV V VI | | | | | PESO | | PUNTATORE 1 |
| 1 2 3 4 5 6 | 1 2 3 4 5 6 | | | | | K | | |
| B R M V A G | B R M V A G | | | | | | | |

```
K     EQU 44000
ADDC  EQU K
X     EQU K+6
O+X   EQU K+7
PUNT  EQU K+8
ADDCP EQU K+10
```

## MASTER MIND

ORG 0 37000, 4385
ORG 0 43000, $40
MX

### SORGENTE

| | | |
|---|---|---|
| 37000 - 3688 | CP Prog Princ. Bytes = 114 | ORG 2, ORG 3, ORG 4, ORG 5 |
| 37200 - 3150 | PUNTATORE Bytes = 184 | ADDCP |
| 37400 | POSIZIONE COLORI Bytes = 946 | ADDCP |
| 38400 | N°COLORI LOW Bytes = 265 | ADDCP |
| 38680 | N°COLORI HIGH Bytes = 301 | ADDCP |
| 39000 | TEST1 N=X Bytes = 159 | IN: IX=CP, IY=C; OUT C=N°X |
| 39200 | TEST2 N°O+X Bytes = 533 | IN: IX=CP, IY=C; DE=0, E=OR OUT A=O+X |
| 39800 | CPR Bytes = 436 | ADDCP, C,X,O+X, ORG6, ORG7, IN: (C), (X), (O+X) |
| 40300 | SIMULA Bytes = 576 | ADDCP, X, O+X, ORG6, ORG7 |
| 40950 | TROVA C Bytes = 272 | ADDCP, C |
| 41250 | CLS PESI Bytes = 138 | |

### OGGETTO

| | |
|---|---|
| 43000 - A7E8 | ORG 1, Bytes = 13 |
| 43012 - A804 | |
| 43020 - A80C | ORG 2, Bytes = 32 |
| 43051 - A82B | |
| 43070 - A83E | ORG 3, Bytes = 183 |
| 43258 - A8FA | |
| 43270 - A906 | ORG 4, Bytes = 45 |
| 43314 - A932 | |
| 43320 - A938 | ORG 5, Bytes = 55 |
| 43374 - A96E | |
| 43376 - A970 | ORG 6, Bytes = 35 |
| 43401 - A991 | |
| 43420 - A99C | ORG X, Bytes = 113 |
| 43532 - AA0C | |
| 43580 - AA3C | ORG 8, Bytes = 74 |
| 43653 - AA85 | |
| 43680 - AAA0 | ORG 9, Bytes = 112 |
| 43791 - AB0F | |
| 43850 - AB6A | ORG 10, Bytes = 53 |
| 43910 - AB86 | ORG 11, Bytes = 23 |

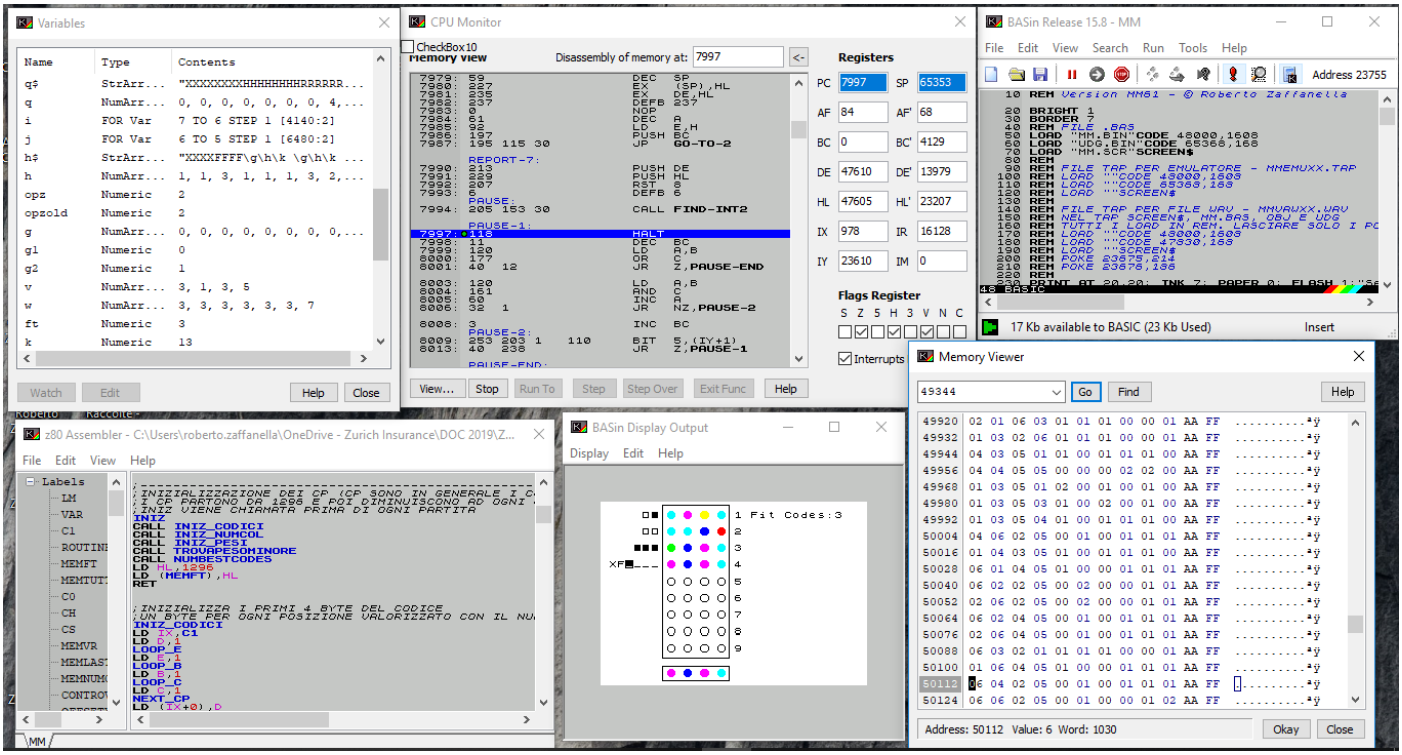
### FILE

| | | |
|---|---|---|
| 44000 - ABE0 | START | C |
| 44006 - ABEC | START+1 | X |
| 44007 - ABED | START+2 | O+X |
| 44008 - ABEE | START+8 | PUNT 0 |
| 44010 - ABF0 | START+10 | CP 1 |
| 44018 - ABF2 | START+18 | PUNT 1 |
| 44020 - ABF4 | CP 2 | |
| 44028 | PUNT 2 | |
| 44030 | CP 3 | |
| 44038 | PUNT 3 | |
| 56950 - DE76 | CP 1295 | |
| 56958 - DE76 | PUNT 1295 | |
| 56960 - DE80 | CP 1296 | |
| 56968 - DE88 | FFXX | |

### MEMORY MAPPING

| | |
|---|---|
| 23000 | CHAMP |
| 36592 | |
| 37000 | MASTER MIND SOURCE |
| 42800 | |
| 43000 | MASTER MIND CODE |
| 43890 | |
| 44000 | MASTER MIND FILE |
| 56968 | |
| 57344 - E000 | ZEUS |
| 65279 | |

La ruggine era molta e ci ho messo un bel po' per riprendere confidenza con l'Assembler e per capire la logica delle routine che avevo scritto. Ripercorrere i ragionamenti fatti 35 anni prima è stato emozionante come rivedere delle foto di quei tempi.
Poi ho cercato di ricrearmi l'ambiente che avevo sullo Spectrum ma questa volta con l'emulatore FUSE e Zeus Assembler + CHAMP in versione TZX; Zeus per programmare e CHAMP per il debugging.
Dopo un po' mi sono reso conto che questo modo di procedere era troppo laborioso e lento e così ho cercato, e per fortuna trovato, un IDE (Integrated Development Environment) da utilizzare su PC Windows chiamato "BASIN".
Il resto è storia recente di tante notti passate a scrivere e soprattutto riscrivere, migliorare, velocizzare il codice e farcelo stare tutto nei 48K.
Grazie veramente agli autori di BasIn. Senza questo tool credo non mi sarebbe stato possibile terminare il progetto.
Mi sento di consigliarlo vivamente a chi vuole sviluppare sia in Basic che Assembler.
Nel caso però consiglio la versione 15.8. Con le versioni successive ho riscontrato dei problemi.
Qui il link per il download:
https://sites.google.com/site/ulaplus/home/zx-spin-and-basin 15.8
Si scompatta in una cartella e non necessità di installazione.
Di seguito, uno screenshot del programma all'opera. Si possono vedere i listati in Basic e Assembler, il programma in esecuzione, il valore delle variabili, il contenuto della memoria e l'esecuzione del codice oggetto:




Un'ultima nota sulla scelta di usare l'inglese per i menu e le altre parti del programma.
Lo spazio video dello Spectrum è molto limitato paragonato a quello a cui siamo abituati da tempo e l'inglese ben si presta per economizzare lo spazio video. Inoltre, il programma può utilizzato da persone non italiane con più facilità.

**Domande, segnalazioni di bug o suggerimenti sono benvenuti. Scrivere a zaffaroby@gmail.com**

**Buon divertimento** 😉