



# ASM FOR ALL CHALLENGE

## Retro-Computer-Entertainment #07 - ASM for all Challenge 2

Inizio 15.04.2020

Terminé 14.05.2020

---

## Secondo appuntamento doppia sfida

In questo secondo appuntamento vi propongo due mini programmi da emulare.

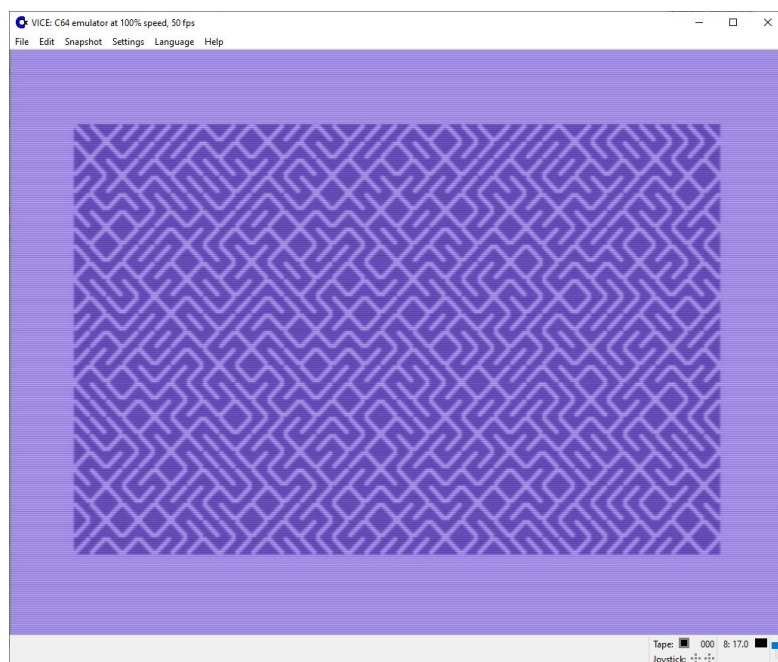
La prima sfida è tratta da una delle più famose righe di Basic del c-64 (ovviamente dopo 10?"PIPP0":GOTO10).

Essa va a riempire lo schermo di caratteri scelti casualmente da un set di due caratteri "\", in questa maniera viene creata una trama che ricorda un labirinto. La sfida naturalmente è quella di riprodurre il comportamento in assembler.

```
1for i=1024to2023:pokei,77.5+rnd(1):next:wait198,1
```

La riga è stata scritta per c-64 in Basic V2, ma è talmente semplice che può essere riprodotta in qualsiasi altro basic (se si vuole vederne il funzionamento in Basic prima della traduzione in ASM).

La riga non fa altro che ciclare su tutte le locazioni di memoria video, nel c64 di default sono dalla 1024 alla 2023 e per ognuna di esse sceglie casualmente tra uno dei due caratteri grafici che mostrano una riga diagonale '/' o '\'.  
Alla fine (non è fondamentale riprodurre questo comportamento) attende la pressione di un tasto (nel c64 il Wait 198,1 svolge questo compito).



Nell'immagine il risultato finale che si vuole ottenere.

## Seconda challenge (Mosquito)

La seconda sfida di questo mese deriva sempre da una unica riga di Basic e anch'essa abbisogna di scelte casuali.

```
0a$="{up}{right}{down}{left}":print"{left}{reverse on} "
mid$(a$,rnd(1)*4+1,1)"{left}{reverse on}*"::fori=1to38:next:goto0
```

Anche in questo caso il basic usato è quello del C-64.

Le scritte tra parentesi graffe in realtà identificano un meta-carattere che l'interprete V2 considera non come un carattere stampabile ma una azione da intraprendere (la sintassi è quella del CBM Prg Studio). Per esempio {Up} indica di spostare il cursore nella cella sopra alla posizione attuale del cursore stesso, le altre direzioni sono {down}{left} e {right}, mentre {reverse on} indica di stampare il carattere successivo con l'inversione di sfondo.

Nel programma viene inizializzata una stringa con i 4 caratteri meta-comandi corrispondenti alle 4 direzioni in cui un moschino (un carattere \*) si può spostare. Prima di spostarsi nella nuova posizione viene cancellato il moschino dalla posizione corrente con uno spazio in inversione di sfondo, in modo da dare l'impressione che il moschino voglia cancellare lo schermo.

Viene scelta casualmente una delle 4 caselle adiacenti, quindi viene stampato il moschino con inversione di sfondo e viene ripetuta la cosa all'infinito.

In aggiunta potete anche provare a usare l'intorno di Moore (8 caselle adiacenti) invece dell'intorno di Von Neumann (4 caselle).

A fianco l'immagine del moschino che cancella lo schermo casualmente.



## Gestione della sfida

Verrà richiesto lo sviluppo di un programma in assembler (qualsiasi assembler supportato da computer 8/16 bit escluse le architetture 8086/8088). Verrà fornito il programma nel formato di poche righe in basic e/o un video che mostra il comportamento che dovrà avere il programma da fare.

I programmi inizialmente saranno molto semplici ma si complicheranno e/o andranno a coprire nuovi aspetti della programmazione assembly con l'andar delle sfide.

Coloro che raccoglieranno la sfida potranno coprire una o tutte le seguenti categorie:

- Sviluppo del programma con le stesse funzionalità;
- Sviluppo del programma con le stesse funzionalità e minor numero di istruzioni a design-time possibili;
- Sviluppo del programma con le stesse funzionalità e maggior velocità di esecuzione.

Naturalmente ognuna delle categorie va considerata all'interno della propria architettura.

## Regolamento

Nello sviluppare il codice, si chiede di evitare di utilizzare eventuali routine presenti in ROM su macchine specifiche (per esempio su C64 le routine del Kernal).

Si chiede di commentare bene il codice, un buon commento fa risparmiare il tempo che si perderebbe a capire cosa faccia un programma fatto da qualcun altro o peggio da se stessi.

Durante il periodo della sfida si incoraggia lo scambio di opinioni e pezzi di codice, anche in prospettiva del fatto che alcune persone si stanno approcciando a questo tipo di programmazione. Si chiede comunque di non pubblicare sin da subito il programma completo, sia per dar modo di imparare a chi non conosce perfettamente l'argomento sia per dare un piccolo tocco competitivo alla sfida.

I programmi Verranno raccolti e catalogati alla fine della sfida.

La durata della sfida è di 1 Mese, quindi se la data del documento di lancio della sfida riporta il giorno 15 (come in questo caso), la sfida terminerà il 14 del mese successivo (compreso).

I programmi potranno essere inviati a [Emanuele.Bonin71@Gmail.Com](mailto:Emanuele.Bonin71@Gmail.Com) accompagnati anche da due righe di spiegazione, sulla metodologia utilizzata o i trucchi utilizzati.