

Sistemi Operativi – a.a. 2025/2025

prova di laboratorio
– 17 luglio 2025 –

Creare un programma **semi-magic-squares-screener.c** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

semi-magic-squares-screener <M-square-size> <bin-file-1> ... <bin-file-N>

Il programma prende in input $N \geq 1$ **file binari** contenenti la rappresentazione di matrici di numeri quadrate $M \times M$, con $3 \leq M \leq 16$, e seleziona quelli che rappresentano "**quadrati semi-magici**". Una matrice $M \times M$ si dice essere un quadrato semi-magico se tutte le sue righe e colonne (ma non le diagonali) sono formate da numeri la cui somma coincide con lo stesso totale (detto "totale semi-magico").

In particolare al suo avvio il programma creerà $N + 1$ thread:

- **N thread lettori** che si occuperanno, rispettivamente, di leggere uno dei file indicati sulla riga di comando; ogni file è formato da record binari di $M \times M$ byte rappresentanti gli elementi di ogni riga della matrice; sono disponibili dei file di esempio: [semi-squares-bin.zip](#); il file dovrà essere letto tramite la **mappatura dei file** in memoria (`mmap`); qualunque altro metodo sarà considerato errato
- **1 thread verificatore** che si occupa di verificare se una matrice $M \times M$ data corrisponde o meno ad un quadrato semi-magico

Le strutture dati condivise saranno:

- una **coda intermedia** di record con capienza massima di 5 elementi che conterrà i quadrati letti dai file da verificare
- un **record finale**, contenente esattamente una sola matrice, usato per passare i quadrati semi-magici verificati al thread principale
- **mutex** e **variabili condizione**: il loro numero, comunque minimale, e le modalità d'uso sono da determinare da parte dello studente
- eventuali flag/contatori di fine-lavoro.

Tutti i thread lettori, agendo in parallelo, si occuperanno di leggere i quadrati descritti nei rispettivi file: per ogni quadrato da verificare creerà un record da inserire nella coda intermedia.

Il thread verificatore si occuperà di estrarre, uno alla volta, i quadrati candidati dalla coda intermedia e di verificarne la natura semi-magica; se viene individuato un quadrato semi-magico, questo sarà inserito nel record finale.

Il thread principale/padre si occuperà di prelevare e visualizzare i quadrati semi-magici inseriti nel record finale dal verificatore.

Tutti i thread dovranno terminare spontaneamente alla fine dei lavori e non si dovranno usare strutture dati con visibilità globale. E' necessario rispettare fedelmente la struttura dell'output riportato nell'esempio a seguire. Codici sorgente con errori che bloccano la compilazione e pregiudicano la generazione del codice binario con compilatori standard (gcc o clang) non saranno valutati.

Tempo: 2 ore e 30 minuti

La struttura dell'output atteso è la seguente:

```
$ ./semi-magic-squares-screener 5 5x5-matrix-sample-a.bin 5x5-matrix-sample-b.bin

[MAIN] creazione di 2 thread lettori e 1 thread verificatore
[READER-2] file '5x5-matrix-sample-b.bin'
[READER-1] file '5x5-matrix-sample-a.bin'
[READER-2] quadrato candidato n.1: (85, 90, 39, 39, 91) (51, 76, 44, 24, 76) (78, 51, 54, 12, 25) (65, 21, 23, 36, 80) (91, 34, 46, 50, 84)
[READER-2] quadrato candidato n.2: (62, 7, 33, 1, 97) (9, 15, 79, 73, 93) (55, 46, 55, 2, 55) (7, 77, 7, 9, 47) (97, 80, 42, 39, 61)
[READER-2] quadrato candidato n.3: (58, 24, 83, 60, 74) (40, 76, 40, 36, 6) (14, 83, 23, 63, 30) (25, 58, 61, 2, 80) (28, 46, 48, 87, 10)
[READER-2] quadrato candidato n.4: (31, 82, 71, 65, 88) (51, 23, 27, 12, 14) (75, 85, 88, 23, 31) (1, 73, 62, 35, 39) (11, 75, 37, 1, 4)
[VERIF] verifico quadrato: (85, 90, 39, 39, 91) (51, 76, 44, 24, 76) (78, 51, 54, 12, 25) (65, 21, 23, 36, 80) (91, 34, 46, 50, 84)
[VERIF] verifico quadrato: (62, 7, 33, 1, 97) (9, 15, 79, 73, 93) (55, 46, 55, 2, 55) (7, 77, 7, 9, 47) (97, 80, 42, 39, 61)
[VERIF] verifico quadrato: (58, 24, 83, 60, 74) (40, 76, 40, 36, 6) (14, 83, 23, 63, 30) (25, 58, 61, 2, 80) (28, 46, 48, 87, 10)
[READER-1] quadrato candidato n.2: (21, 35, 70, 21, 87) (60, 35, 89, 34, 90) (69, 62, 14, 83, 6) (32, 15, 24, 21, 35) (71, 56, 1, 13, 34)
...
[VERIF] verifico quadrato: (87, 16, 55, 49, 71) (22, 48, 86, 62, 60) (59, 57, 38, 86, 38) (99, 99, 12, 37, 31) (11, 58, 87, 44, 78)
[VERIF] trovato quadrato semi-magico!
...
[READER-2] quadrato candidato n.17: (16, 66, 24, 54, 8) (16, 93, 33, 38, 9) (38, 53, 28, 86, 77) (86, 74, 61, 41, 11) (10, 72, 45, 27, 96)
[MAIN] quadrato semi-magico trovato:
  (87, 16, 55, 49, 71)
  (22, 48, 86, 62, 60)
  (59, 57, 38, 86, 38)
  (99, 99, 12, 37, 31)
  (11, 58, 87, 44, 78)
  totale semi-magico 278
[VERIF] verifico quadrato: (4, 84, 45, 97, 98) (80, 90, 80, 75, 27) (18, 4, 26, 97, 11) (91, 12, 68, 17, 70) (9, 72, 88, 93, 40)
...
[READER-2] quadrato candidato n.50: (87, 45, 58, 93, 71) (75, 53, 39, 38, 43) (29, 15, 8, 30, 50) (3, 89, 33, 77, 52) (5, 97, 86, 49, 93)
[READER-2] terminazione con 50 quadrati letti
[VERIF] verifico quadrato: (87, 68, 59, 16, 84) (17, 59, 65, 85, 53) (28, 58, 86, 4, 32) (35, 16, 39, 46, 84) (10, 2, 25, 52, 93)
[VERIF] verifico quadrato: (87, 45, 58, 93, 71) (75, 53, 39, 38, 43) (29, 15, 8, 30, 50) (3, 89, 33, 77, 52) (5, 97, 86, 49, 93)
[VERIF] terminazione
[MAIN] terminazione con 10 quadrati semi-magici trovati
```