

Sistemi Operativi – a.a. 2025/2025

prova di laboratorio
– 1 luglio 2025 –

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

magic-squares-screener <M-verifiers> <file-1> ... <file-N>

Il programma prende in input N file testuali contenenti la descrizione di matrici di numeri quadrate 3x3 e seleziona quelli che rappresentano "quadrati magici". Una matrice 3x3 si dice essere un quadrato magico se tutte le sue righe (comprese le diagonali principali) sono formate da numeri la cui somma coincide con lo stesso totale (detto "totale magico"). Ecco alcuni esempi di quadrati magici:

8	1	6
3	5	7
4	9	2

totale: 15

16	2	12
6	10	14
8	18	4

totale: 30

24	3	18
9	15	21
12	27	6

totale: 45

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

- N thread lettori che si occuperanno, rispettivamente, di leggere uno dei file indicati sulla riga di comando; ogni riga del file rappresenta una matrice di interi 3x3 nel formato "36,13,27,9,24,41,18,3,30"; sono disponibili dei file di esempio: [squares-1.txt](#) [squares-2.txt](#) [squares-3.txt](#)
- M thread verificatori si occuperanno di verificare se una matrice 3x3 data corrisponde o meno ad un quadrato magico

Le strutture dati condivise saranno:

- una coda intermedia di record con capienza massima di 10 elementi che conterrà i quadrati letti dai file da verificare
- una coda finale di record con capienza massima di 3 elementi che conterrà solo quadrati magici verificati
- 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.

Tutti i thread verificatori, agendo sempre in parallelo, si occuperanno di estrarre, una alla volta, i quadrati candidati dalla coda intermedia e di verificarne la natura magica; se viene individuato un quadrato magico, questo sarà inserito nella coda finale.

Il thread principale/padre si occuperà di estrarre e visualizzare i quadrati magici inseriti nella coda finale.

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 15 minuti

La struttura dell'output atteso è la seguente:

```
$ ./magic-squares-screener 2 squares-1.txt squares-2.txt squares-3.txt

[MAIN] creazione di 3 thread lettori e 2 thread verificatori
[READER-1] file 'squares-1.txt'
[READER-1] quadrato candidato n.1: (3, 4, 10) (40, 9, 15) (10, 6, 11)
[READER-3] file 'squares-3.txt'
[READER-3] quadrato candidato n.1: (1, 15, 35) (42, 24, 13) (7, 47, 27)
[READER-1] quadrato candidato n.2: (12, 48, 7) (33, 1, 25) (40, 19, 5)
[READER-2] file 'squares-2.txt'
[READER-2] quadrato candidato n.1: (50, 21, 3) (14, 29, 41) (8, 33, 17)
[READER-3] quadrato candidato n.2: (5, 42, 18) (33, 9, 21) (50, 2, 14)
[VERIF-1] verifico quadrato: (3, 4, 10) (40, 9, 15) (10, 6, 11)
[VERIF-2] verifico quadrato: (1, 15, 35) (42, 24, 13) (7, 47, 27)
[READER-3] quadrato candidato n.3: (27, 1, 38) (41, 15, 6) (20, 34, 49)
[READER-2] quadrato candidato n.2: (6, 36, 23) (49, 12, 4) (31, 20, 44)
[READER-1] quadrato candidato n.3: (8, 1, 6) (3, 5, 7) (4, 9, 2)
[READER-3] quadrato candidato n.4: (10, 44, 25) (3, 19, 39) (47, 8, 30)
[VERIF-1] verifico quadrato: (12, 48, 7) (33, 1, 25) (40, 19, 5)
[VERIF-2] verifico quadrato: (27, 1, 38) (41, 15, 6) (20, 34, 49)
[VERIF-2] verifico quadrato: (6, 36, 23) (49, 12, 4) (31, 20, 44)
[VERIF-1] verifico quadrato: (8, 1, 6) (3, 5, 7) (4, 9, 2)
...
[VERIF-1] trovato quadrato magico!
[VERIF-2] verifico quadrato: (10, 44, 25) (3, 19, 39) (47, 8, 30)
[READER-2] quadrato candidato n.5: (7, 23, 40) (14, 28, 34) (43, 11, 26)
[READER-1] quadrato candidato n.6: (4, 25, 39) (44, 19, 2) (17, 31, 47)
[VERIF-1] trovato quadrato magico!
[VERIF-2] verifico quadrato: (29, 1, 41) (13, 33, 8) (20, 48, 15)
[MAIN] quadrato magico trovato:
  (8, 1, 6)
  (3, 5, 7)
  (4, 9, 2)
  totale 15
[READER-2] quadrato candidato n.4: (36, 12, 22) (5, 46, 24) (10, 3, 50)
[MAIN] quadrato magico trovato:
  (16, 2, 12)
  (6, 10, 14)
  (8, 18, 4)
  totale 30
...
[VERIF-2] verifico quadrato: (22, 1, 16) (7, 13, 19) (10, 25, 4)
[VERIF-2] trovato quadrato magico!
[READER-2] terminazione
[VERIF-1] terminazione
[VERIF-2] terminazione
[MAIN] quadrato magico trovato:
  (22, 1, 16)
  (7, 13, 19)
  (10, 25, 4)
  totale 39
[MAIN] terminazione
```