Sistemi Operativi – a.a. 2024/2025
prova di laboratorio
– 17 settembre 2025 –

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

`equisum-array-screener <file-bin-1> <file-bin-2> ... <file-bin-`*`N`*`>`

Il programma prende in input **N≥1 file binari** contenenti la rappresentazione di alcuni **vettori** con **M=12 interi** ciascuno in cui ogni elemento è rappresentato da un **byte senza segno**. Lo stesso dovrà individuare all'interno delle sequenze di vettori, quelli che godono della **proprietà di equisomma**: un vettore di interi si dice "equisomma" se la somma dei numeri nelle posizioni pari è uguale alla somma dei numeri nelle posizioni dispari. Esempio:

30	20	10	15	5	25	40	10	20	5	25	55

(somma posizioni pari = somma posizioni dispari = 130)

In particolare al suo avvio il programma creerà **N + 3 thread** ausiliari:
- **N thread lettori** che si occuperanno, rispettivamente, di leggere uno dei file indicati sulla riga di comando; ogni file è formato da un certo numero di record binari di M=12 byte rappresentanti gli elementi di un vettore; il file dovrà essere letto tramite la **mappatura dei file in memoria** (`mmap`); qualunque altro metodo sarà considerato errato
- **3 thread verificatori** che si occuperanno di verificare se un dato vettore goda o meno della proprietà di equisomma.

Le strutture dati condivise saranno:
- una **coda intermedia** di record con capienza massima di **10 elementi** che conterrà i vettori da verificare letti dai file in input
- un **record finale**, contenente esattamente **un vettore**, usato per passare i vettori equisomma verificati al thread principale
- **mutex** e **semafori POSIX**: 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 vettori descritti nei rispettivi file: per ogni vettore da verificare creerà un record da inserire nella coda intermedia.
I thread verificatori, anch'essi agendo in parallelo, si occuperanno di estrarre i vettori candidati dalla coda intermedia e di verificarne la natura; se viene individuato un vettore equisomma, questo sarà inserito nel record finale.
Il thread principale/padre si occuperà di prelevare e visualizzare i vettori equisomma inseriti nel record finale dai verificatori.

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.

Suggerimento: per prevenire la miscelazione degli output dei vari thread su più comandi di stampa (`printf`) è sempre possibile impiegare, con le dovute attenzioni, `flockfile`/`funlockfile` su `stdout`.

Tempo: 2 ore e 30 minuti

La struttura dell'output atteso sui file di esempio dati: `vectors-A.bin`, `vectors-B.bin`, `vectors-C.bin`

```
$ ./equisum-array-screener vectors-A.bin vectors-B.bin vectors-C.bin

[MAIN] creazione di 3 thread lettori e 3 thread verificatori
[READER-1] file 'vectors-A.bin'
[READER-2] file 'vectors-B.bin'
[READER-1] vettore candidato n.1: 181, 213, 18, 67, 115, 13, 17, 162, 145, 6, 31, 191
[READER-1] vettore candidato n.2: 182, 101, 230, 192, 73, 243, 113, 68, 118, 170, 141, 83
[READER-3] file 'vectors-C.bin'
[READER-2] vettore candidato n.1: 109, 95, 141, 172, 192, 239, 143, 41, 49, 40, 150, 108
[READER-2] vettore candidato n.2: 122, 127, 205, 132, 212, 117, 103, 81, 62, 225, 139, 161
[VERIF-1] verifico vettore: 181, 213, 18, 67, 115, 13, 17, 162, 145, 6, 31, 191
[VERIF-1] non è un vettore equisomma (somma pari 507 vs. dispari 652)
[READER-1] vettore candidato n.3: 134, 113, 120, 66, 177, 214, 55, 214, 0, 251, 107, 195
[VERIF-1] verifico vettore: 182, 101, 230, 192, 73, 243, 113, 68, 118, 170, 141, 83
[VERIF-1] si tratta di un vettore equisomma con somma 857!
[MAIN] ricevuto nuovo vettore equisomma: 182, 101, 230, 192, 73, 243, 113, 68, 118, 170, 141,
83
[VERIF-3] verifico vettore: 109, 95, 141, 172, 192, 239, 143, 41, 49, 40, 150, 108
[VERIF-3] non è un vettore equisomma (somma pari 784 vs. dispari 695)
...
[READER-2] terminazione con 50 vettori letti
[VERIF-3] terminazione con 78 vettori verificati
[READER-3] terminazione con 80 vettori letti
[VERIF-1] verifico vettore: 59, 86, 77, 95, 25, 41, 170, 14, 161, 231, 127, 180
[VERIF-1] non è un vettore equisomma (somma pari 647 vs. dispari 619)
[READER-1] terminazione con 100 vettori letti
[VERIF-1] terminazione con 73 vettori verificati
[VERIF-2] terminazione con 79 vettori verificati
[MAIN] ricevuto nuovo vettore equisomma: 182, 23, 217, 181, 114, 192, 244, 160, 177, 246, 45,
177
[MAIN] terminazione con 30 vettori equisomma trovati
```