

# Informatica industriale LT

## Prova scritta – 2 febbraio 2026 – 2h (1h)

NOTA per chi svolge la seconda prova parziale

- Se si svolge la seconda prova parziale, si deve rispondere solamente alle domande 1, 4, 5, 6, 9
- Se si svolge la seconda prova parziale, si ha 1 ora di tempo, e non 2

### PARTE 1 – RISPOSTA SINGOLA

Ogni domanda ha una sola risposta VERA.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
- Una risposta lasciata in bianco viene valutata 0

1. (3, -.5) La memoria *heap*:

- a) E' quella in cui vengono allocate le strutture dinamiche, usando ad esempio API come *malloc* o *new*
- b) E' quella in cui vengono allocate le strutture statiche, ad esempio gli array
- c) E' quella in cui è presente il codice macchina del programma (segmento *text*)
- d) E' quella in cui finiscono gli oggetti quando "svuoto" il cestino

2. (3, -.5) La programmazione con Functional Block Diagrams:

- a) E' un paradigma di programmazione PLC che non prevede mai l'uso di Ladder
- b) E' un paradigma di programmazione PLC che non prevede mai l'uso di Structured Text
- c) E' un paradigma di programmazione PLC che non prevede mai l'uso di counter up (CTU)
- d) Nessuna delle precedenti

3. (3, -.5) Un sistema a *polling*:

- a) E' più reattivo di un sistema asincrono, ad esempio *event-driven*, ma occupa più risorse computazionali
- b) E' meno reattivo di un sistema asincrono, ad esempio *event-driven*, ma occupa meno risorse computazionali
- c) E' più reattivo di un sistema asincrono, ad esempio *event-driven*, ma occupa meno risorse computazionali
- d) E' meno reattivo di un sistema asincrono, ad esempio *event-driven*, ma occupa più risorse computazionali

## PARTE 2 - (POSSIBILI) RISPOSTE MULTIPLE -

Ogni domanda può avere da zero a quattro risposte CORRETTE.

- Ogni risposta esatta viene calcolata: +1
  - Ogni risposta errata viene calcolata: -0.5
  - Una risposta lasciata in bianco viene calcolata: 0
4. Perché è utile implementare un sistema concorrente (competitivo o collaborativo)?
- a) Perché senza un sistema competitivo non si può mai usare la GPGPU
  - b) Perché entrambi consentono di utilizzare al meglio il tempo di stallo di processi o thread riallocando i core disponibili
  - c) Perché aumentano implicitamente la sicurezza informatica del systems
  - d) Non è vero. Non sono utili, ma sono uno stile di programmazione obsoleto e che ormai esiste solo nei sistemi *legacy*
5. Il pattern a Finite State Machine:
- a) Non è applicabile nei sistemi embedded
  - b) E' applicabile nei sistemi embedded, se non hanno la GPGPU
  - c) E' applicabile nei sistemi embedded ma non consente di usare altri pattern
  - d) E' applicabile nei sistemi embedded solamente se girano su un sistema operativo di alto livello come GNU/Linux
6. La shared memory della GPU:
- a) Fisicamente è la stessa della L1
  - b) Non è gestibile dal programmatore
  - c) Serve a far comunicare i blocchi
  - d) E' condivisa tra più thread dello stesso blocco
7. In un paradigma collaborativo, o multi-thread:
- a) Non sono mai possibili le data race
  - b) Sono possibili le data race, e non c'è modo di gestirle
  - c) Sono possibili le data race, e si riesce a gestirle grazie all'uso di lock
  - d) Per evitare le data race occorre saper riconoscere le *critical section*
8. I Field Programmable Gate Arrays:
- a) Possono essere programmati con *High-Level Synthesis*
  - b) Non ha senso che vengano usati nell'industria moderna
  - c) Sono collegabili come coprocessore ad un host core, implementando il paradigma eterogeneo
  - d) Non riescono a supportare l'implementazione di carichi computazionali di AI

### PARTE 3 – DOMANDE APERTE

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore
- Una risposta lasciata in bianco viene calcolata: 0
- L'eventuale sfioramento del limite di righe o parole (laddove imposto), porterà a una decurtazione di un punto per ogni riga. Eventuali schematici e listati di codice non verranno presi in considerazione nel calcolo delle righe
- **SI RICORDA CHE L'UNICO FOGLIO DA CONSEGNARE E' IN CALCE AL COMPITO. QUESTO FOGLIO PUO' SERVIRE ESCLUSIVAMENTE COME "BRUTTA COPIA". EVENTUALI RISPOSTE SCRITTE IN QUESTO FOGLIO NON VERRANNO PRESE IN CONSIDERAZIONE**

9. **(9 pt)** Descrivere il flusso di lavoro tipico dell'host e del device di quando viene eseguito del codice sulla GPU. Descrivere anche come il programmatore può gestire queste fasi da codice.

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

10. **(5 pt)** Si descrivano i punti salienti dello standard POSIX Threads (PThreads) in 5 righe

# Informatica industriale LT

## Prova scritta – 2 febbraio 2026 – 2h

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare.

**NOTA per chi svolge la seconda prova parziale**

- Se si svolge la seconda prova parziale, si deve rispondere solamente alle domande 1, 4, 5, 6, 9
- Se si svolge la seconda prova parziale, si ha 1 ora di tempo, e non 2

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					3	-0.5
4						
5						
6						
7						
8						

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

**Risposta alla domanda 9 (9 pt):**

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

**Risposta alla domanda 10 (5 pt):**