PCAD a.a. 2022/2023
<DATA>

L'esame è composto da 10 domande a risposta multipla e 2 a risposta libera.
Nelle prime 10 domande bisogna indicare se le affermazioni sono vere o false.
Se avete dubbi sulla formulazione della domanda aggiungete una breve spiegazione per giustificare la risposta.
Nella stessa domanda ci possono essere zero o più affermazioni vere.

D1 (1 punto) Quando si utilizza un semaforo in un programma concorrente
1. i thread non possono accedere simultaneamente alla propria sezione critica
2. l'accesso a dati condivisi da parte di più thread viene serializzato
3. il primo thread che accede al semaforo ha priorità su tutti gli altri nelle istruzioni seguenti
4. è opportuno inizializzare il semaforo a 1 se si vuole usare come mutex

D2 (1 punto) La libreria CyclicBarrier di Java
1. Viene usata per controllare parallelismo tra thread
2. Viene usata come alternativa alle Memory Fence nei modelli con weak consistency
3. Ha un metodo "wait" che viene usato per sincronizzare thread
4. Ha un metodo "notify" che viene usato per sbloccare thread in attesa

D3 (1 punto) La tecnica di programmazione lock-free
1. viene usata per minimizzare il numero di istruzioni tra acquire e release di un lock
2. viene usata per rendere efficiente la gestione delle race condition nei programmi concorrenti
3. viene usata per favorire e facilitare l'uso di lock e mutex nei programmi concorrenti
4. viene solo usata nell'implementazione di strutture dati concorrenti

D4 (1 punto) Un Mutex
1. è un semaforo che può assumere solo i valori occupato e non occupato
2. è un oggetto immutabile in Java
3. non può essere usato come campo di una struttura dati tipo lista o array
4. può essere implementato disabilitando interruzioni

D5 (1 punto) Un ReadWrite lock
1. è un semaforo binario che viene usato su oggetti con metodi getter e setter
2. viene usato per garantire la mutua esclusione tra thread che aggiornano una variabile condivisa
3. viene usato per garantire la mutua esclusione tra thread che leggono una variabile condivisa
4. garantisce starvation-freedom se usato per controllare una risorsa condivisa

D6 (2 punti) In un programma concorrente

1. un errore su un certo input si può riprodurre ripetendo l'esecuzione una sola volta
2. la funzione calcolata dal programma associa ad un certo input un insieme di output
3. con lo stesso input posso avere un'esecuzione che termina e una che non termina
4. con lo stesso input posso avere un'esecuzione che termina e una che si blocca

D7 (2 punti) Quando usiamo oggetti callable in Java
1. Le chiamate dei metodi corrispondenti possono restituire valori
2. Le chiamate dei metodi corrispondenti sono effettuate in mutua esclusione
3. Le chiamate dei metodi corrispondenti sono tutte effettuate in maniera asincrona
4. Non possiamo propagare le eccezioni al di fuori dei metodi corrispondenti

D8 (2 punti) Nel contesto degli iteratori in Java
1. sono generalmente oggetti Thread-Safe
2. possono essere utilizzati solo in mutua esclusione
3. possono sollevare eccezione ConcurrentModificationException solo quando uno (o più) thread modificano l'oggetto associato all'iteratore
4. possono sollevare eccezione ConcurrentModificationException anche quando Weakly Consistent

D9 (2 punti) L'assenza di race-condition
1. si ottiene dichiarando thread che non allocano dati sullo heap
2. si ottiene creando thread che allocano memoria solo attraverso variabili locali
3. si ottiene garantendo che i dati siano letti o modificati solo all'interno di singoli thread
4. si ottiene acquisendo lock su dati condivisi tra diversi thread

D10 (2 punti) Nell'esecuzione di un programma concorrente
1. tutti i thread lanciati da un programma vengono sempre eseguiti almeno per un'istruzione
2. i thread vengono eseguiti in parallelo quando possibile ma non necessariamente
3. non è possibile avere due context-switch consecutivi dello stesso thread
4. il numero di context-switch dipende dal numero di interrupt sollevate