# Calcolo parallelo LT
# Esempio di prova scritta –2h

PARTE 1 – RISPOSTA SINGOLA - Ogni domanda ha una sola risposta VERA, indicata in rosso.

- 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) OpenMP è
   a) Una interfaccia di programmazione di basso livello
   b) Una interfaccia di programmazione di alto livello
   c) Una interfaccia di programmazione client-server
   d) Un linguaggio di programmazione nato per sopperire alle carenze del C, e pertanto incompatibile con esso

2. (3, -.5) I loop OpenMP
   a) Possono essere statici o dinamici, a seconda di come vengono allocati i chunk di lavoro ai vari thread
   b) Possono essere statici o dinamici, a seconda di come viene suddiviso in chunk il lavoro che verrà poi allocato ai vari thread
   c) Possono essere statici o dinamici, a seconda di come vengono creati i chunk di lavoro agendo sulla clausola private
   d) Possono essere statici o dinamici, a seconda di come vengono creati i chunk di lavoro agendo sulla clausola shared

3. (3, -.5) La API di programmazione CUDA
   a) Consente di definire lo spazio dei thread, su cui poi verrà mappato lo spazio di lavoro o dei dati
   b) Consente di definire thread, thread groups e thread linking
   c) E' stata creata da un consorzio di cui fanno parte diversi produttori di GPU. Fra gli altri, NVIDIA, ATI e RADEON
   d) Non è portabile attraverso le diverse famiglie di GPU NVIDIA

## PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE -
## Ogni domanda può avere da una a quattro risposte CORRETTE, indicate in rosso.

- Ogni risposta esatta viene calcolata: +1
- Ogni risposta errata viene calcolata: -0.5
- Una risposta lasciata in bianco viene calcolata: 0

4. Un semaforo è
   a) Un costrutto che consente la sincronizzazione fra più thread concorrenti
   b) Un costrutto che può essere utilizzato esclusivamente a livello di sistema operativo, per ragioni di protezione della memoria
   c) Inizializzabile solo con valori positivi
   d) Inizializzabile anche a zero, e in quel caso rimane "bloccato"

5. Il costrutto lock
   a) E' presente solo in OpenMP
   b) E' presente in diverse API di programmazione, anche se talvolta con altri nomi
   c) Implementa un meccanismo di basso livello
   d) Può essere implementato con apposito hardware dedicato, come le memorie *test-and-set*

6. OpenMP e CUDA sono simili nel senso che
   a) Consentono un controllo a grana molto fine dei thread e della memoria
   b) Consentono di implementare il *data parallelism* (o parallelismo di dati), ognuno attraverso appositi costrutti di linguaggio
   c) Prevedono costrutti di barriera, implicita o esplicita
   d) Non supportano parallelismo irregolare o dinamico

7. Un dato in cache/memoria temporanea si dice coerente
   a) Quando, anche per caso, assume lo stesso valore del dato presente in memoria centrale
   b) Quando viene automaticamente aggiornato con il dato presente in memoria centrale, se quest'ultimo è modificato da qualche altro thread
   c) Quando una sua modifica viene automaticamente propagata in memoria centrale attraverso un meccanismo di *write-and-lock*
   d) Quando la cache svuotata nella cache di un altro core della piattaforma

8. Un processo
   a) E' un programma in esecuzione
   b) E' un programma appena prima di venire eseguito
   c) Ha uno spazio di memoria condiviso con gli altri processi
   d) Può comunicare con gli altri processi attraverso primitive MPI

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 sforamento del limite di righe o parole (laddove imposto), porterà a una decurtazione di un punto per ogni riga

9. (5 pt) Si descriva in non più di 15 righe i possibili meccanismi per la distribuzione del lavoro che si sono visti durante il corso. Se lo si vuole, si può citare alcuni linguaggi tipo OpenMP o CUDA, ma non è indispensaibile. (un aiuto/soluzione schematica è in rosso, sotto)

10. (6 pt) Il seguente snippet di codice contiene un errore di concetto, che ne preclude il corretto funzionamento, sebbene in teoria il codice compili.

Si trovi l'errore, e si proponga una soluzione

```c
int main()
{
  int i;
  float a[16], b[16];

  // a e b vengono inizializzati in qualche modo

  #pragma omp parallel for num_threads(4) shared(i)
  for(i=0; i<16; i++)
    b[i] = a[i];
}
```