Esercitazione 5

1. Eseguire il programma somma_tempi (disponibile sulla piattaforma e-learning), usando tre diverse configurazioni del kernel, nel rispetto delle limitazioni imposte dalla compute capability della GPU utilizzata (per Google Colaboratory occorre verificare prima del lancio del programma).
   Calcolare i tempi di esecuzione e lo Speed up al variare della dimensione del problema N (N≥100·000).

2. Implementare in linguaggio CUDA-C un programma che calcoli la somma di due matrici quadrate di dimensione N.
   - Memorizzare le matrici in array monodimensionali*.
   - Configurare il kernel come una griglia bidimensionale di BxB blocchi, con blocchi bidimensionali di TxT threads, con tre diversi valori di T.

   Per ogni configurazione del kernel usata, calcolare il numero di blocchi residenti in uno streaming multiprocessor e il numero di thread attivi, in base alla GPU utilizzata.
   Eseguire il programma sviluppato, usando le tre diverse configurazioni del kernel.
   Calcolare i tempi di esecuzione e lo Speed up, al variare della dimensione N del problema, con N>1·000. Utilizzare valori di N multipli di 32, ad esempio N=1·024, 2048, 4·096, 8·1962, 16·384, … .
3. **Facoltativo.** Svolgere l'esercizio 2, considerando matrici rettangolari N x M, con griglie e blocchi rettangolari, cioè una griglia bidimensionale di (Bx,By) blocchi, con blocchi bidimensionali di (Tx,Ty) threads. Utilizzare una sola configurazione per il kernel, che sia ottimale rispetto alla compute capability della GPU utilizzata.

*Si potrebbero usare array 2D, ma la gestione sarebbe più complessa e si dovrebbero utilizzare specifiche funzioni CUDA