

Enunciados Ejercicios.

Sesión 1:

Ejercicio 1: Escribe y ejecuta un programa en CUDA (N bloques de 1 thread) que lea un vector de N reales en la GPU y obtenga como resultado un vector de tamaño N-2 que contenga la media de tres elementos consecutivos del vector de entrada: ejemplo: vector entrada=[0 1 2 3 4 5 6 7 8 9] vector salida =[1 2 3 4 5 6 7 8]

Archivo resuelto **ej_sesion1_mediavector.cu**

Ejercicio 2: Escribe y ejecuta un programa en CUDA que lea una matriz M*N en la gpu y devuelva un vector con N componentes que contenga las medias de las columnas de la matriz.

(Hacerlo con un thread por cada columna de la matriz, 1 bloque de N threads), Ejemplo,

matriz $\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix}$ vector salida =[1,2,3,4]

Archivo resuelto **ej_sesiones_1_2_mediamatrizv1_completo.cu**

Sesión 2:

Ejercicio 1: A partir del programa de la sesión anterior para sumar dos vectores, amplíalo a una versión bidimensional que sume dos matrices de la misma dimensión. Cada thread debe sumar un solo elemento. Por ejemplo, tomar un grid de bloques como el del ejemplo (grid 2 por 3, bloques 4 por 4), con una matriz 8 por 12.

Archivo resuelto **ej-sesion_2_sumamatriz.cu**

Ejercicio 2: Queremos implementar en la gpu un filtro para suavizar una imagen (matriz). Dada una matriz Inp de dimensiones M filas por N columnas, queremos calcular una matriz Out de dimensiones M-2 filas por N-2 columnas , de forma que el elemento [i,j] de Out se calcule así:

$$\text{Out}(i,j)=(\text{Inp}(i-1,j)+\text{Inp}(i+1,j)+\text{Inp}(i,j-1)+\text{Inp}(i,j+1)+\text{Inp}(i,j))/5.0$$

De forma que cada thread haga el cálculo de un solo elemento de la matriz Out. Hazlo con un bloque bidimensional.

Archivo resuelto **ej_sesion2_filtro.cu**

Sesión 3

Ejercicio 1: Acaba de implementar el ejercicio de producto escalar de dos vectores (Puedes partir del archivo en poliformat, prod_esc_incompleto.c).

Archivo resuelto **ej_sesion_3_prod_esc_completo.cu**

Ejercicio 2: Haz un programa para calcular en la GPU las medias de las columnas de una matriz usando reducción. Cada bloque calcula la media de una columna, haciendo colaborar todos los threads del bloque. Hazlo a partir del archivo en poliformat `media_matriz_incompleto.c`

Archivo resuelto **`ej_sesion_3_mediamatrizv2_completo.cu`**

Ejercicio 3: El objetivo del ejercicio es escribir un programa en CUDA que calcule el número total de veces que una cierta secuencia de longitud 2 (por ejemplo 6,3) que aparecen en una matriz cuadrada de números enteros entre 0 y 9 (Problema relacionado con secuenciación de ADN). Las secuencias se deben buscar por columnas.

Tenéis en poliformat un archivo, `contar_int_incompleto_col.cu`, en el que: 1) hay una función, llamada “`contar_int`”, que calcula en la CPU el número de apariciones de una cierta secuencia de longitud 2. Hay otra función para imprimir la matriz. 2) El main genera la memoria en la CPU, genera una matriz de números enteros aleatorios, la imprime y llama a la función `contar_int`, mostrando por pantalla el resultado. Debéis completar ese código con lo necesario para que se calcule en la gpu el número de apariciones de la secuencia (que por supuesto debe coincidir con el número calculado en la CPU).

Para facilitar el trabajo, se pueden entregar dos versiones, de orden creciente de dificultad: - una básica, versión 1, en la que se usa un thread o bloque por columna, y solo se buscan las secuencias por columnas. -versión 2, en la que usa un bloque por columna y varios threads colaboran para contar las secuencias en su columna. Como en los ejemplos vistos en clase, se tiene que hacer una última reducción en la CPU.

Archivo resuelto: **`ej_sesion_3_busca_int_completo_col.cu`** (incluye las dos versiones)