King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No 6bis – Quarter 3 2023

Question

1. Let's consider the following serial nested loops:

```
for (int i = 0 ; i < N; i ++)
      for (int j = 0 ; j < M; i ++)
            doSomething(i, j);
```

   a. Give a parallel solution using CUDA C.

2. We would like to write a C program that calculates in parallel the matrix-vector product.

   Let's consider a ***N by N*** square matrix. Let's consider a vector (1-D array) of size ***N***. The matrix-vector product is calculated as follows:

$$\begin{matrix} \boldsymbol{a_0^0} & \dots & \boldsymbol{a_0^{n-1}} \\ \dots & \dots & \dots \\ \boldsymbol{a_{n-1}^0} & \dots & \boldsymbol{a_{n-1}^{n-1}} \end{matrix} \times \begin{matrix} b_0 \\ \dots \\ b_{n-1} \end{matrix} = \begin{matrix} \sum_{j=0}^{n-1} a_0^j \times b_j \\ \dots \\ \sum_{j=0}^{n-1} a_{n-1}^j \times b_j \end{matrix}$$

   So, we would like to write a kernel that receives 4 parameters:

   - a matrix of integers denoted **A** (**input**),
   - a vector of integers denoted **B** (**input**),
   - a vector of integers, denoted **C** (**output**), where the matrix-vector product will be stored. Elements of vector C will be calculated as follows:

$$C[i] = \sum_{j=1}^{N} \mathrm{A[i][j]} * \mathrm{B[j]}$$

   - an integer denoted **N** (**input**) which is the size of the vector B; and the number of rows and columns of the matrix A.

a. We would like to run this kernel within a grid composed of multiple blocks each of which is vector of threads. We would like that every thread calculates a single cell of the result (**the vector C**). Give the code of the kernel.