**GPU**

Q1. Modify sum.cu to implement polynomial evaluation function given an array of coefficients and a value x. Submit the CUDA program and a screenshot that shows you ran the code on Mill.

For example:

int\* coefficients = {11, 12, 13}.

This array means 11 \* x^0  +  12 \* x^1  +  13 \* x^2

Output Array = {11, 12, 13}   (if x = 1)

Final Output value for a polynomial = 11+ 12+13 = 36

Here is a sequential program to demonstrate the polynomial evaluation code. You have to develop a CUDA version of this code and run it on Mill GPU. The input array coeffArr contains the coefficients of a large polynomial explicitly and the degree information of the terms implicitly. Each output term of the polynomial is stored in the outputTerms array which should be copied back to the CPU from the GPU. CPU should then add the terms to get the final output value for a polynomial.

void evaluate( int x,  int []coeffArr, int []outputTerms, int COEFFICIENTS)   
{  
        int maxDegree = COEFFICIENTS - 1;  
        for( int term = 0; term < COEFFICIENTS; term++ )

        {

              int coefficient = coeffArr[ term ];

              int result = Math.pow(x, term);    // term = degree

              outputTerms[ term ] = coefficient\*result;  
        }

}

Note: Addition of the terms to get the final answer is not shown in the method above.