A1

Dibyadarshan Hota (16CO154)
Omkar Prabhu (16CO233)

Q1) Write a CUDA program to calculate the element-wise sum of two large (16K elements or greater) arrays. The arrays contain single precision floating point numbers. Generate your input arrays.
A1)

Initialization of input arrays

```
48      // initialization of host arrays
49      for(int i = 0; i < ARRAY_SIZE; ++i) {
50          host_arrA[i] = i + 3;
51          host_arrB[i] = 2*i + 1;
52      }
```

Output: First 10 values

```
PS P:\heterogeneous-parallel-computing\A1> ./a
Input A:
3.000000 4.000000 5.000000 6.000000 7.000000 8.000000 9.000000 10.000000 11.000000 12.000000
Input B:
1.000000 3.000000 5.000000 7.000000 9.000000 11.000000 13.000000 15.000000 17.000000 19.000000
Result:
4.000000 7.000000 10.000000 13.000000 16.000000 19.000000 22.000000 25.000000 28.000000 31.000000
```

Q2) Perform Matrix Addition of two large **integer matrices** in CUDA. Answer the following questions.
A2)

Initialization of matrix

```
46      // initialization of host arrays
47      for(i=0; i<ROW_SIZE; i++) {
48          for(j=0; j<COL_SIZE; j++) {
49              host_matrix1[i][j] = i;
50              host_matrix2[i][j] = j;
51              host_matrix3[i][j] = 0;
52          }
53      }
```

Output: For first 25 values

```
PS P:\heterogeneous-parallel-computing\A1> nvcc .\q2.cu
q2.cu
   Creating library a.lib and object a.exp
PS P:\heterogeneous-parallel-computing\A1> ./a
Result:
0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
3 4 5 6 7
4 5 6 7 8
PS P:\heterogeneous-parallel-computing\A1>
```

1. How many floating operations are being performed in the matrix addition kernel?
A1) Since the matrices are integer matrices (mentioned in the question), the number of float operations are 0.

2. How many global memory reads are being performed by your kernel?
A2) Since 3 matrices are being read (3 loads), the number of reads are:
=> 3 * row_size * col_size = 3 * 1024 * 1024 = 3,145,728

3. How many global memory writes are being performed by your kernel?
A3) Since 1 matrix is being assigned values, the number of writes are:
=> row_size * col_size = 1024 * 1024 = 1,048,756