VECTOR ADDITION ON GPU

Submitted by : Goli Uma Sankar
Regd. No. : 18560

The overall structure of a CUDA program that uses the GPU for computation is as follows:

- Define the code that will run on the device in a separate function, called the *kernel* function, i.e., **add()** function given below :-

```
__global__ void add(long long int *a, long long int *b, long long int *c)
{
    long long int index = threadIdx.x + blockIdx.x * blockDim.x;

    c[index] = a[index] + b[index];
}
```

- In the main program running on the host's CPU:-
  a) allocate memory on the host for the data arrays.
  b) initialze the data arrays in the host's memory.
  c) allocate separate memory on the GPU device for the data arrays.
  d) copy data arrays from the host memory to the GPU device memory.

- On the GPU device, execute the **add()** that computes new data values given the original arrays. Specify how many blocks and threads per block to use for this computation as :-

```
add<<< ceil(N/THREADS_PER_BLOCK), THREADS_PER_BLOCK >>>(d_a, d_b, d_c);
```

- After the **add()** function completes, copy the computed values from the GPU device memory back to the host's memory.