

```
#define N 8
```

```
__global__ void add(int *a, int *b, int *c) {  
    int tid = blockDim.x * blockIdx.x + threadIdx.x;  
  
    while (tid < N) {  
        c[tid] = a[tid] + b[tid];  
        tid += blockDim.x;  
    }  
}  
  
int main(){  
    int *dev_a, *dev_b, *dev_c;  
  
    //Memory allocation and initializing  
  
    //Kernel initializing  
    init_kernel_listing();  
    create_kernel(std::get<0>(get_listing()), add, gridSize, blockSize, dev_a, dev_b, dev_c);  
  
    //Periodic task parameters  
    struct gpu_sched_param ga;  
    ga.period_us = 5000;  
    ga.deadline_us= 5000;  
    ga.priority = 3;  
  
    //Periodic task creation  
    struct pruda_task_t * p_task_a = create_pruda_task(0, ga, gridSize, blockSize);  
  
    //Using Fixed Priority algorithm with the Single-stream strategy  
    init_scheduler(SINGLE, FP);  
    add_pruda_task(p_task_a);  
  
    //One CPU thread per CUDA kernel (see pruda_task() function example above)  
    create_cpu_threads();  
  
    //CPU operations  
  
    //Memory free  
  
    return 0;  
}
```
