```
#include<stdio.h>

__global__ void add(int* a, int* b, int* c, int n)
{
      int idx = threadIdx.x;
      if(idx<n)
      c[idx] = a[idx]+ b[idx];
}

int main(){
      int n;
      scanf("%d",&n);
      float elapsed_time;
      cudaEvent_t start,stop;
      cudaEventCreate(&start);
      cudaEventCreate(&stop);
      cudaEventRecord(start,0);

      cudaStream_t stream0,stream1;
      cudaStreamCreate(&stream0);
      cudaStreamCreate(&stream1);


      int *h_a,*h_b,*h_c;
      cudaHostAlloc((void**)&h_a,20*n*sizeof(int),cudaHostAllocDefault);
      cudaHostAlloc((void**)&h_b,20*n*sizeof(int),cudaHostAllocDefault);
      cudaHostAlloc((void**)&h_c,20*n*sizeof(int),cudaHostAllocDefault);

      for(int i=0; i<20*n; i++){
            h_a[i]=i;
            h_b[i]=i+1;
      }

      for(int i=0; i<20*n; i+=n){
            int *d_a,*d_b,*d_c;
            cudaMalloc((void**)&d_a,n*sizeof(int));
            cudaMalloc((void**)&d_b,n*sizeof(int));
            cudaMalloc((void**)&d_c,n*sizeof(int));
            int seg1=n*(7/10);
            int seg2=n-seg1;

            cudaMemcpyAsync(d_a,h_a+i,
seg1*sizeof(int),cudaMemcpyHostToDevice,stream0);
            cudaMemcpyAsync(d_b,h_b+i,
seg1*sizeof(int),cudaMemcpyHostToDevice,stream0);
            cudaMemcpyAsync(d_a,h_a+i+seg1,
seg2*sizeof(int),cudaMemcpyHostToDevice,stream1);
            cudaMemcpyAsync(d_b,h_b+i+seg1,
seg2*sizeof(int),cudaMemcpyHostToDevice,stream1);

            add<<<1,seg1,0,stream0>>>(d_a,d_b,d_c,seg1);
            add<<<1,seg2,0,stream1>>>(d_a,d_b,d_c,seg2);


      cudaMemcpyAsync(h_c+i,d_c,seg1*sizeof(int),cudaMemcpyDeviceToHost,s
tream0);
```

```
      cudaMemcpyAsync(h_c+i+seg1,d_c,seg2*sizeof(int),cudaMemcpyDeviceToH
ost,stream1);

            cudaFree(d_a);
            cudaFree(d_b);
            cudaFree(d_c);
      }

            cudaStreamSynchronize(stream0);
            cudaStreamSynchronize(stream1);
            cudaEventRecord(stop,0);
            cudaEventSynchronize(stop);
            cudaEventElapsedTime(&elapsed_time,start,stop);
            printf("Time:%3.1f\n",elapsed_time);

            for(int i=0; i<20*n; i++)
                  printf("%d ",h_c[i]);
            cudaFreeHost(h_a);
            cudaFreeHost(h_b);
            cudaFreeHost(h_c);

            cudaEventDestroy(stop);
            cudaEventDestroy(start);


            cudaStreamDestroy(stream0);
            cudaStreamDestroy(stream1);
            return 0;
}
```