```
#include<stdio.h>
#include<stdlib.h>
__global__ void matadd(int *d_a,int *d_b,int *d_c, int n){

int idx=threadIdx.x;
if(idx<n)
d_c[idx]=d_a[idx]+d_b[idx];

}


int main(){

int n;
scanf("%d",&n);
cudaEvent_t start,stop;
float escap_time;


cudaEventCreate(&start);
cudaEventCreate(&stop);

cudaEventRecord(start,0);

cudaStream_t stream;
cudaStreamCreate(&stream);




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;
}

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));


for(int i=0; i<20*n; i+=n){
      cudaMemcpyAsync(d_a,h_a+i,n*sizeof(int),cudaMemcpyHostToDevice,stre
am);
      cudaMemcpyAsync(d_b,h_b+i,n*sizeof(int),cudaMemcpyHostToDevice,stre
am);

matadd<<<1,n,0,stream>>>(d_a,d_b,d_c,n);

      cudaMemcpyAsync(h_c+i,d_c,n*sizeof(int),cudaMemcpyDeviceToHost,stre
am);
}
cudaStreamSynchronize(stream);
cudaEventRecord(stop,0);
cudaEventSynchronize(stop);
cudaEventElapsedTime(&escap_time,start,stop);
printf("Time:%3.1f\n",escap_time);

for(int i=0; i<20*n; i++)
 printf("%d ",h_c[i]);
cudaFreeHost(h_a);
cudaFreeHost(h_b);
cudaFreeHost(h_c);
cudaEventDestroy(start);
cudaEventDestroy(stop);
cudaFree(h_a);
cudaFree(h_b);
cudaFree(h_c);
cudaStreamDestroy(stream);
return 0;

}
```