

Результаты:

```
!./task1
```

Generated N=1000000 random floats. First 10 values:

h[0]=0.242676

h[1]=0.760386

h[2]=0.735512

h[3]=0.916327

h[4]=0.886008

h[5]=0.146871

h[6]=0.167343

h[7]=0.630454

h[8]=0.516711

h[9]=0.308807

[12]

✓ 0s

```
!./task2_global 1000000
```

▼

Reduce (global atomic) N=1000000

GPU sum=500374 | CPU sum=500374

Time: 3.5223 ms

[13]

✓ 0s

```
!./task2_shared 1000000
```

▼

Reduce (shared+global) N=1000000

GPU sum=500124 | CPU sum=500124

Time: 0.221152 ms

[14]

✓ 0s



```
!./task3_bubble 100000
```

▼

... Bubble sort local subarrays: N=100000, SUB=32

First 10 subarrays sorted? YES

```
[16]
✓ 0s
!./task3_merge 100000

Merge shared pass: N=100000, width=128
Each merged segment sorted? YES
```

```
[19]
✓ 3s
!nvcc -O2 -arch=sm_75 task4_performance_measure.cu -o task4
!./task4

... Task4 performance
BLOCK_SIZE=256, SUB_SIZE=32

N      reduce_global(ms)  reduce_shared(ms)  bubble(ms)  merge(ms)
10000  0.208864               0.027744           0.049696   0.036896
100000 0.350176               0.020384           0.179904   0.059392
1000000 3.45389               0.101568           1.65856    0.48128

Saved CSV: performance.csv
```

Performance.csv:

	N ↑	Reduce_global ↓	Reduce_shared ↓	Bubble_subarray ↓	Merge_pairs ↓
	10000	0.21	0.03	0.05	0.04
	100000	0.35	0.02	0.18	0.06
	1000000	3.45	0.1	1.66	0.48

Ответы на контрольные вопросы:

#### 1. Типы памяти CUDA

- Глобальная — большая, медленная
- Разделяемая — быстрая, на блок
- Локальная — регистры, на поток

#### 2. Влияние shared memory

Снижает обращения к глобальной памяти → ускорение в разы.

#### 3. Доступ и его обеспечение

Использование coalesced-доступа и синхронизации `__syncthreads()`.

#### 4. Сложности больших данных

Ограничение памяти, пропускная способность, латентность.

#### 5. Минимизация глобальной памяти

Глобальная память — главный узкий ресурс GPU.

## 6. Профилирование

Используются nvprof, Nsight Compute, Nsight Systems.