

!Пояснения к оцениванию заданий!

- на оценку **Удовлетворительно** - достаточно выполнить первое задание;
- на оценку **Хорошо** - выполнить или первое и второе (MPI), или первое и третье (MPI + CUDA) задания;
- на оценку **Отлично** - выполнить первое, третье и четвертое задания.

1 Задание первое

1. Реализовать программу, в которой каждый процесс печатает число процессов в группе и свой номер в ней;
2. Реализовать при помощи послыки сообщений типа точка-точка следующие схемы коммуникации процессов:
 - передача данных по кольцу, два варианта: «эстафетная палочка» (очередной процесс дожидается сообщения от предыдущего и потом посылает следующему) и «сдвиг» (одновременные посылка и прием сообщений);
 - master-slave (все процессы общаются с одним);
 - пересылка данных от каждого процесса каждому.

2 Задание второе

Сортировка слиянием (задание из слайдов лекций):

1. Реализовать последовательные версии сортировки слиянием;
2. Реализовать параллельные версии сортировки слиянием;
3. Сгенерировать наборы входных данных (случайные массивы целых чисел, ≥ 10 одного размера, ≥ 5 градаций размера). Размеры подобрать в зависимости от системы.
4. Провести запуск сортировок на данных, замерить время исполнения.

Задание третье

Задание 1.1. Сложение двух векторов

Написать программу, которая преобразует последовательный код, находящийся в папке "Содержимое/CUDA/Код для заданий" в код на CUDA C с параметрами ядра $\lll 1, 1 \ggg$. Затем изменить параметры ядра $\lll N, 1 \ggg$ (*blockIdx.x*) и $\lll 1, N \ggg$ (*threadIdx.x*).

Задание 1.2

Изменить задание 1.1, увеличив количество элементов в векторах и применить распараллеливание на блоки и нити

```
<<< (Num_elements + (Num_threads - 1))/Num_threads , Num_threads >>>  
  
(blockIdx.x * blockDim.x + threadIdx.x).
```

Задание четвёртое. Работа с матрицами

1. Написать программу сложения двух матриц, используя двумерную нумерацию сетки и блоков:

```
dim3 blocks(Num_elements/Num_threads_x, Num_elements/Num_threads_y);  
dim3 threads(Num_threads_x, Num_threads_y);
```

2. Написать программу для умножения матриц, используя разделяемую память и синхронизацию.