

Онлайн-компилятор:

<https://www.codingame.com/playgrounds/349/introduction-to-mpi/hello-world>

## **Задачи:**

(прочитать спецификации задач на странице 2)

1. Hello world из всех процессов (2 балла) - 1
2. Максимум массива (3 балла) - 2
3. Вычисление числа Пи методом Монте-Карло (3 баллов) - 2
4. Среднее арифметическое среди положительных чисел массива (3 баллов) - 2
5. Скалярное произведение векторов (3 балла) - 2
6. Maxmin матрицы (5 балла) - 3
7. Умножение матрицы на вектор при разделении данных по столбцам (7 баллов - doc7.pdf) - 4
8. Scatter и Gather через Send и Recv(6 баллов) - 3
9. Инвертировать массив (6 баллов) - 3
10. Время передачи для разных Send-ов (6 баллов) - 3
11. Циклическая передача данных (6 баллов) - 3

*Итого 1 - 11 = 50 баллов*

12. Передача чисел по кругу для различных коммуникаторов(+ 3 балла)
13. Проверка матрицы на симметричность (9 баллов)
14. Сортировка массива методом чет-нечетной перестановки (13 баллов - doc10.pdf)

### **На отлично:**

15. Умножение матриц алгоритм Кэннона (15 баллов - doc8.pdf)
16. Умножение матриц алгоритм Фокса (15 баллов - doc8.pdf)
17. Сортировка данных методом Шелла (15 баллов - doc10.pdf)
18. Метод Гаусса (15 баллов - doc9.pdf)

19. Быстрая сортировка с выбором ведущего элемента (15 баллов - doc10.pdf)
20. Алгоритм Флойда-Уоршелла (15 баллов - doc11.pdf)
21. Алгоритм Краскала (15 баллов - doc11.pdf)
22. Алгоритм Прима (15 баллов - doc11.pdf)
23. Алгоритм Кернигана – Лина (15 баллов)

### **Спецификации задач:**

- 1 – уметь запускать MPI
- 2 – уметь использовать что-то из send, recv, bcast, reduce
- 3 – метод Монте-Карло для получения  $\Pi$ , понимается как генерирование случайной последовательности точек из квадрата со сторонами 2 и центром в центре координат. Доля точек попавших в круг с радиусом один умноженная на 4 должна стремиться к числу  $\Pi$ . Распараллеливание заключается в сбалансированном распределении итераций по процессам.
- 4 - Распределение данных через scatterv. При сборе данных использовать одну операцию но с двумя числами.
- 5 - Распределение данных через scatterv. Есть два массива – надо найти суммы произведений соответствующих координат.
- 6 – Найти седловую точку матрицы. Распределение данных через scatterv. Каждый процесс получает какое-то количество строк матрицы. Находит для каждой строки минимум и выбирает из них максимальный. Далее из локальных максимумов выбирает глобальный. Проверить совпадают ли maxmin и minmax.
- 7 - На одном процессе заполняется матрица и вектор. Каждый процесс получает несколько столбцов матрицы и столько же элементов вектора. Рассчитывает частичную сумму результирующего вектора.  $c_i = A_{i,j} * b_j$  для j-го столбца. Если столбцов несколько, то  $c_j$  суммируются. Далее все частичные суммы собираются в результат.
- 8 – На одном процессе есть массив из n чисел. Выводим его. При помощи send, recv раздаем всем процессам по n/size чисел. Свою часть так же копируем в другой массив размера n/size. Выводим номер каждого процесса и его часть массива. Далее при помощи send, recv собираем все части массива на каком-либо процессе в новый массив размера n. Выводим его.
- 9 – Перевернуть массив. Работа процессов должна быть сбалансирована. Можно использовать как и Scatterv, Gatherv, так и Send, Recv.

10 – Программу тестируем на двух процессах. Используем Send, Ssend, Bsend и Rsend - передаем какой-либо длинный массив или строку второму процессу, и получаем обратно. Замеряем время потраченные на эти операции.

11 – В коммуникаторе передаем сообщение от одного процесса другому с нулевого до size-1. Последний процесс отправляет сообщение нулевому. На каждом процессе сообщение изменяется, если это число, то можно прибавлять или умножать на что-то.

12 - После полного круга из задачи 11, создаем новый коммуникатор (произвольным образом, с меньшим количеством процессов) и повторяем процедуру.

13 – Каждый процесс должен получить только нужные ему данные, работа процессов должна быть сбалансирована.

14 - 23 задачи есть в pdf файлах. Рассмотрим алгоритмы на ближайших занятиях.