**Task 2**

1. В каждом подчиненном процессе дано целое число. Переслать эти числа в главный процесс, используя функции MPI\_Send и MPI\_Recv, и вывести их в главном процессе. Полученные числа выводить в порядке возрастания рангов переславших их процессов.
2. В каждом подчиненном процессе даны четыре целых числа. Переслать эти числа в главный процесс, используя по одному вызову функции MPI\_Send для каждого передающего процесса, и вывести их в главном процессе. Полученные числа выводить в порядке возрастания рангов переславших их процессов.
3. В каждом подчиненном процессе дано целое число N (0 < N < 5) и набор из N целых чисел. Переслать данные наборы в главный процесс, используя по одному вызову функции MPI\_Send для каждого передающего процесса, и вывести наборы в главном процессе в порядке возрастания рангов переславших их процессов. Для определения размера пересланного набора использовать функцию MPI\_Get\_count.
4. В главном процессе дан набор вещественных чисел; количество чисел равно количеству подчиненных процессов. С помощью функцииMPI\_Send переслать по одному числу в каждый из подчиненных процессов (первое число в процесс 1, второе — в процесс 2, и т. д.) и вывести в подчиненных процессах полученные числа.
5. В главном процессе дан набор вещественных чисел; количество чисел равно количеству подчиненных процессов. С помощью функции MPI\_Send переслать по одному числу в каждый из подчиненных процессов, перебирая процессы в обратном порядке (первое число в последний процесс, второе — в предпоследний процесс, и т. д.), и вывести в подчиненных процессах полученные числа.
6. В главном процессе дано целое число N и набор из N чисел; K − 1 ≤ N < 10, где K — количество процессов. С помощью функции MPI\_Send переслать по одному числу их данного набора в процессы 1, 2, …, K − 2, а оставшиеся числа — в процесс K − 1, и вывести полученные числа. В процессе K − 1 для определения количества полученных чисел использовать функцию MPI\_Get\_count.
7. В каждом подчиненном процессе дано целое число, причем только для одного процесса это число отлично от нуля. Переслать ненулевое число в главный процесс и вывести в главном процессе полученное число и ранг процесса, переславшего это число. Для приема сообщения в главном процессе использовать функцию MPI\_Recv с параметром MPI\_ANY\_SOURCE.
8. В каждом подчиненном процессе дано целое число N, причем для одного процесса это число больше нуля, а для остальных равно нулю. В процессе с ненулевым N дан также набор из N чисел. Переслать данный набор чисел в главный процесс и вывести в главном процессе полученные числа и ранг процесса, переславшего этот набор. При приеме сообщения использовать параметр MPI\_ANY\_SOURCE.
9. В каждом подчиненном процессе дано целое число N, в главном процессе дано целое число K (> 0), равное количеству тех подчиненных процессов, в которых даны положительные числа N. Переслать все положительные числа N в главный процесс и вывести в нем сумму полученных чисел. Для приема сообщений в главном процессе использовать функцию MPI\_Recv с параметром MPI\_ANY\_SOURCE.
10. В каждом процессе дано вещественное число. Переслать число из главного процесса во все подчиненные процессы, а все числа из подчиненных процессов — в главный, и вывести в каждом процессе полученные числа (в главном процессе числа выводить в порядке возрастания рангов переславших их процессов).
11. В каждом процессе дано целое число. С помощью функций MPI\_Send и MPI\_Recv осуществить для всех процессов циклический сдвиг данных с шагом 1, переслав число из процесса 0 в процесс 1, из процесса 1 в процесс 2, …, из последнего процесса в процесс 0. В каждом процессе вывести полученное число.
12. В каждом процессе дано целое число. С помощью функций MPI\_Send и MPI\_Recv осуществить для всех процессов циклический сдвиг данных с шагом −1, переслав число из процесса 1 в процесс 0, из процесса 2 в процесс 1, …, из процесса 0 в последний процесс. В каждом процессе вывести полученное число.

**Task 3**

1. В главном процессе дано целое число. Используя функцию MPI\_Bcast, переслать это число во все подчиненные процессы и вывести в них полученное число
2. В каждом процессе даны два числа: вещественное A и целое N, причем набор чисел N содержит все значения от 0 до K − 1, где K — количество процессов. Используя функции MPI\_Send и MPI\_Recv (с параметром MPI\_ANY\_SOURCE), выполнить в каждом процессе пересылку числа A в процесс N и вывести полученное число, а также ранг процесса, из которого число было получено.
3. В главном процессе дан набор из 5 чисел. Используя функцию MPI\_Bcast, переслать этот набор во все подчиненные процессы и вывести в них полученные числа в том же порядке
4. В каждом процессе дано целое число N, причем для одного процесса значение N равно 1, а для остальных равно 0. В процессе с N = 1 дан также набор из K − 1 числа, где K — количество процессов. Переслать из этого процесса по одному из чисел данного набора в остальные процессы, перебирая ранги получателей в возрастающем порядке, и вывести в каждом из них полученное число.
5. В каждом процессе дан набор из K − 1 целого числа, где K — количество процессов. Для каждого процесса переслать по одному из данных в нем чисел в остальные процессы, перебирая ранги процессовполучателей в возрастающем порядке, и вывести полученные числа в порядке возрастания рангов переславших их процессов.
6. Количество процессов — четное число. В каждом процессе дано целое число N (0 < N < 5) и набор из N чисел. С помощью функции MPI\_Sendrecv выполнить обмен исходными наборами между парами процессов 0 и 1, 2 и 3, и т. д. В каждом процессе вывести полученный набор чисел.
7. В каждом процессе дано вещественное число. С помощью функции MPI\_Sendrecv\_replace поменять порядок исходных чисел на обратный (число из процесса 0 должно быть передано в последний процесс, число из процесса 1 — в предпоследний процесс, …, число из последнего процесса — в процесс 0). В каждом процессе вывести полученное число.
8. В каждом подчиненном процессе дано вещественное число A и его порядковый номер N (целое число); набор всех номеров N содержит все целые числа от 1 до K − 1, где K — количество процессов. Переслать числа A в главный процесс и вывести их в порядке, соответствующем возрас-танию их номеров N. Для передачи номера N указывать его в качестве параметра msgtag функции MPI\_Send.
9. В каждом подчиненном процессе дано целое число L (≥ 0) и набор из L пар чисел (A, N), где A — вещественное, а N — его порядковый номер. Все числа L в сумме равны 2K, где K — количество процессов; набор номеров N, данных во всех процессах, содержит все целые числа от 1 до 2K. Переслать числа A в главный процесс и вывести их в порядке, соответствующем возрастанию их номеров N. Для передачи номера N указывать его в качестве параметра msgtag функции MPI\_Send.
10. В главном процессе дан набор пар чисел (T, A); количество пар равно числу подчиненных процессов. Число T — целое, равное 0 или 1. Число A — целое, если T = 0, и вещественное, если T = 1. Переслать по одному числу A в каждый из подчиненных процессов (первое число в процесс 1, второе — в процесс 2, и т. д.) и вывести полученные числа. Для передачи информации о типе пересланного числа указывать число T в качестве параметра msgtag функции MPI\_Send, для получения этой информации использовать функцию MPI\_Probe с параметром MPI\_ANY\_TAG.
11. В каждом подчиненном процессе даны два целых числа T, N и набор из N чисел. Число T равно 0 или 1. Набор содержит целые числа, если T = 0, и вещественные числа, если T = 1. Переслать исходные наборы в главный процесс и вывести полученные числа в порядке возрастания рангов переславших их процессов. Для передачи информации о типе пересланного числа указывать число T в качестве параметра msgtag функции MPI\_Send, для получения этой информации использовать функцию MPI\_Probe с параметром MPI\_ANY\_TAG.
12. Процесс с номером size-1 вводит значение целого типа в перемнную temp. Рассылает это значение остальным процессам. Процессы с четными номерами меняют знак у temp и посылают полученнле значение на процессы с нечетными номерами, большими на единицу . Каждый процесс печатает текущее значение перемнной temp и свой ранг.

Task 4

1. Написать программу для определения примерного времени t, в главном процессе дан набор из 5 чисел. Используя функцию MPI\_Bcast, переслать эти числа во все подчиненные процессы, каждый процессор суммирует полученные значения и умножает на свой номер, и отпраляет обратно в нулевой процессор. Вывести на экран полученные значения по номеру переславшего прцессора. И выполните данную задачу без импользования функции BCAST - используйте функции MPI\_Send и MPI\_Recv и определите время через функцию MPI\_Wtime. Сравните.
2. На нулевом процессе с помощью датчика случайных чисел сформировать из 5 вещественных чисел: x[k]=sin(k\*rand()), k=1,2,…5. С помощью функции Bcast разослать эти значения остальным процессам, вычислить сумму этих чисел на процессе с номером 1, и получить минимальное число на процессе 2. В остальных процессах умножить на номер процессора. И выести полученные значения с номером процессора. Определите затраченное время.
3. На нулевом процессе ввести значение целого типа в переменную temp. Для инициализированных процессов организовать передачу этого значения по схеме 0→1, 0→2, ... 0→size-1 с помощью функции Bcast. Затем каждый процесс умножает temp на свой номер, и полученный результат возвращается на нулевой процесс, где он суммируется с результатами, полученными с остальных процессов. Сумма выдается на каждом номере процессоров. Определите затраченное время.
4. На нулевом процессе с помощью датчика случайных чисел сформировать из 10 вещественных чисел: x[k]=cos(k\*rand()), k=1,2,…10. С помощью функции Bcast разослать эти значения остальным процессам, умножить на каждом процессе элементы на его номер, и и определить максимальный элемент на каждом процессоре, и отправьте эти значения в нулевой процессор.
5. На нулевом процессе с помощью датчика случайных чисел сформировать из 5 вещественных чисел: x[k]=sin(k\*rand()), k=1,2,…5. Найти манимальное и максимальное значение. С помощью функции Bcast разослать эти значения остальным процессам. Процессоры с четным номером вывести максимальное значение. Процессором с нечетным номером вывести минимальное значение. Определите затраченное время. И выполните данную задачу без импользования функции BCAST - используйте функции MPI\_Send и MPI\_Recv и определите время через функцию MPI\_Wtime. Сравните.
6. Создайте и выполните программу, используя коммуникационную функцию (MPI\_Bcast), реализующую следующий алгоритм: на 0-м процессоре считывается целочисленное значение N, рассылается всем процессорам, где на каждом из процессоров считываются значение x = N \*rand(). Определите максимальный элемент на каждом процессоре, и отправьте эти значения в нулевой процессор. Определите затраченное время.
7. На нулевом процессе с помощью датчика случайных чисел сформировать массив из 10 вещественных чисел: x = cos(k rand()) ,k =1,2,...,10 . С помощью процедуры MPI\_BCAST разослать эти значения остальным процессам, умножить на каждом процессе элементы на его номер и определите максимальный элемент на каждом процессоре, и отправьте эти значения в нулевой процессор, в нулевом процессоре найдите максимальный элемент среди всех полученных элементов. Определите затраченное время.
8. На нулевом процессе с помощью датчика случайных чисел сформировать массив из 10 вещественных чисел: x = cos(k rand()) ,k =1,2,...,10 . С помощью процедуры MPI\_BCAST разослать эти значения остальным процессам, умножить на каждом процессе элементы на его номер и определите минимальный элемент на каждом процессоре и отправтье в нулевой процессор. В нулевом процессоре определите самый минимальный элемент из полученных значений. Определите затраченное время.
9. На нулевом процессе с помощью датчика случайных чисел сформировать из 5 вещественных чисел: x[k]=sin(k\*rand()), k=1,2,…5. С помощью функции Bcast разослать эти значения остальным процессам, вычислить сумму этих чисел на процессе с номером 1, и получить минимальное число на процессе 2. В остальных процессах умножить на номер процессора. И выести полученные значения с номером процессора. Определите затраченное время.
10. Каждый процесс с помощью датчика случайных чисел задает переменную temp=cos^(rank+1) (rand()). На нулевом процессе с помощью датчика случайных чисел сформировать из 5 вещественных чисел: x[k]=cos(k\*rand()), k=1,2,…5. С помощью функции Bcast разослать эти значения остальным процессам. Каждый процессор умножает полученные эти значения на переменную temp. Определить на каждом процессоре минимальный элемент . Определите затраченное время.
11. На нулевом процессе с помощью датчика случайных чисел считываются значение

x = N \*rand().С помощью функции Bcast разослать эти значения остальным процессам.

Каждый процесс с помощью датчика случайных чисел задает переменную temp=сos((rank+1)\*rand()), затем просуммировать значения переменной x и temp в переменную Sum на каждом процессоре. Переслать эти значения в нулевой процессор и найти мин и макс значения. Определите затраченное время

1. На нулевом процессе с помощью датчика случайных чисел сформировать массив из 10 вещественных чисел: x = k\*rand(). ,k =1,2,...,10 . С помощью процедуры MPI\_BCAST разослать эти значения остальным процессам, умножить на каждом процессе элементы на его номер и определите максимальный элемент на каждом процессоре, и отправьте эти значения в нулевой процессор, в нулевом процессоре найдите максимальный элемент среди всех полученных элементов. Определите затраченное время.

Task 5

1. В каждом про(цессе дан набор из K + 5 целых чисел, где K — количество процессов. Используя функцию MPI\_Reduce для операции MPI\_SUM, просуммировать элементы данных наборов с одним и тем же порядковым номером и вывести полученные суммы в главном процессе.
2. В каждом процессе дан набор из K + 5 чисел, где K — количество процессов. Используя функцию MPI\_Reduce для операции MPI\_MIN, найти минимальное значение среди элементов данных наборов с одним и тем же порядковым номером и вывести полученные минимумы в главном процессе.
3. В каждом процессе дан набор из K + 5 целых чисел, где K — количество процессов. Используя функцию MPI\_Reduce для операции MPI\_MAXLOC, найти максимальное значение среди элементов данных наборов с одним и тем же порядковым номером и ранг процесса, содержащего это максимальное значение. Вывести в главном процессе вначале все максимумы, а затем — ранги содержащих их процессов.
4. MPI-процесс с номером 0 вводит с клавиатуры массив из 8 целых чисел. Затем с помощью функции MPI\_Bcast рассылает этот массив всем процессам. Каждый процесс печатает полученные данные. Произвести сложение всех элементов массива, распределенных по процессам, с помощью функций MPI\_REDUCE с получением результата на процессе с номером 2. Вычисленное значение распечатать.
5. На каждом из size инициализированных MPI-процессов компоненты массива m y , y ,..., y 1 2 вычисляются по формуле y = e , (k x) k cos \* k =1,2,...m, m = 6 , значение x на процессе с номером rank = 0,1,...size −1 определяется с помощью датчика случайных чисел: x = sin((rank +1)\* rand()). Вычислить минимальное значение среди компонентов массивов y и выдать результат на процессе с номером size-1.
6. На нулевом процессе с помощью датчика случайных чисел сформировать из 10 вещественных чисел: x[k]=cos(k\*rand()), k=1,2,…10. С помощью функции Bcast разослать эти значения остальным процессам, умножить на каждом процессе элементы на его номер, используя функцию MPI\_Reduce определить максимальный элемент на каждом процессоре, и отправьте эти значения в нулевой процессор.
7. Создайте и выполните программу, используя коммуникационную функцию (MPI\_Bcast), реализующую следующий алгоритм: на 0-м процессоре считывается целочисленное значение N, рассылается всем процессорам, где на каждом из процессоров считываются значение x = N \*rand().Используя функцию MPI\_Reduce определите максимальный элемент на каждом процессоре, и отправьте эти значения в нулевой процессор.
8. На нулевом процессе с помощью датчика случайных чисел сформировать массив из 10 вещественных чисел: x = cos(k rand()) ,k =1,2,...,10 . С помощью процедуры MPI\_BCAST разослать эти значения остальным процессам, умножить на каждом процессе элементы на его номер используя функцию MPI\_Reduce определите минимальный элемент на каждом процессоре и отправтье в нулевой процессор.
9. В каждом процессе дан набор из K + 5 целых чисел, где K — количество процессов. Используя функцию MPI\_Reduce для операции MPI\_SUM, просуммировать элементы данных наборов с одним и тем же порядковым номером и вывести полученные суммы в главном процессе.
10. В каждом процессе дан набор из K + 5 чисел, где K — количество процессов. Используя функцию MPI\_Reduce для операции MPI\_MIN, найти минимальное значение среди элементов данных наборов с одним и тем же порядковым номером и вывести полученные минимумы в главном процессе.
11. В каждом процессе дан набор из K + 5 целых чисел, где K — количество процессов. Используя функцию MPI\_Reduce для операции MPI\_MAXLOC, найти максимальное значение среди элементов данных наборов с одним и тем же порядковым номером и ранг процесса, содержащего это максимальное значение. Вывести в главном процессе вначале все максимумы, а затем — ранги содержащих их процессов.
12. MPI-процесс с номером 0 вводит с клавиатуры массив из 8 целых чисел. Затем с помощью функции MPI\_Bcast рассылает этот массив всем процессам. Каждый процесс печатает полученные данные. Произвести сложение всех элементов массива, распределенных по процессам, с помощью функций MPI\_REDUCE с получением результата на процессе с номером 2. Вычисленное значение распечатать.

Task 6

Функция MPI\_Isend MPI\_Irecv

1. Напишите программу, разбивающую число n на простые множители. Для этого сопоставьте каждому процессу по одному простому числу. Передавайте число n по кольцу процессов коммуникатора, проверяя его делимость на соответствующие простые числа и уменьшая (деля) n в случае делимости без остатка. В случае если n не делится на простое число, то передать число n без изменений, иначе передать полученное новое значение.
2. Напишите параллельную программу, выводящую на экран таблицу умножения произвольного размера в виде матрицы. При этом не допускайте путаницы строк. И через Isend и Irecv вывести полученные значения в 0 процессе.
3. В каждом процессе дано вещественное число. С помощью функции MPI\_Isend, MPI\_Irecv поменять порядок исходных чисел на обратный (число из процесса 0 должно быть передано в последний процесс, число из процесса 1 — в предпоследний процесс, …, число из последнего процесса — в процесс 0). В каждом процессе вывести полученное число.
4. В каждом процессе дано целое число. С помощью функций MPI\_Isend, MPI\_Irecv осуществить для всех процессов циклический сдвиг данных с шагом 2, переслав число из процесса 0 в процесс 2, из процесса 2 в процесс 4, …, из последнего процесса в процесс 2. В каждом процессе вывести полученное число.
5. В главном процессе дан набор вещественных чисел; количество чисел равно количеству подчиненных процессов. С помощью функции MPI\_Isend переслать по одному числу в каждый из подчиненных процессов, перебирая процессы в обратном порядке (первое число в последний процесс, второе — в предпоследний процесс, и т. д.), и вывести в подчиненных процессах полученные числа.
6. Напишите программу, где в главном процессе дано число n и сопоставьте каждому процессу по одному случайному числу. Передавайте число n по кольцу процессов коммуникатора, проверяя если число n больше чем число в процессоре, то отнимите от n данное число, и отправьте другому процессору. Если число n меньше чем число в процессоре, то нужно просто передать число n другому. Вывести полученное число n на каждом процессоре.
7. На нулевом процессе ввести значение целого типа в переменную temp. Для инициализированных процессов организовать передачу этого значения по схеме 0→1, 1→2, ... size-2→size-1 с помощью функции MPI\_Isend, MPI\_Irecv . Затем каждый процесс умножает temp на свой номер, и полученный результат возвращается на нулевой процесс. Вывести полученные значения temp от каждого процессора.
8. На нулевом процессе с помощью датчика случайных чисел сформировать из 10 вещественных чисел: x[k]= k\*rand(), k=1,2,…10. С помощью функции MPI\_Isend, MPI\_Irecv эти значения остальным процессам против часовой стрелки 2→1 умножить на каждом 0→size-1, size-1→size-2,…, процессе элементы на его номер, и и отправьте эти значения в нулевой процессор.
9. На нулевом процессе с помощью датчика случайных чисел сформировать из 5 вещественных чисел: x[k]=k\*rand(), k=1,2,…5. Найти манимальное и максимальное значение. С помощью функции MPI\_Isend, MPI\_Irecv разослать процессам с четным номером максимальное значение, а с нечетными номером вывести минимальное значение.
10. В каждом процессе дано вещественное число. С помощью функции MPI\_Isend, MPI\_Irecv поменять порядок исходных чисел на обратный (число из процесса 0 должно быть передано в последний процесс, число из процесса 1 — в предпоследний процесс, …, число из последнего процесса — в процесс 0). В каждом процессе вывести полученное число.
11. В 0 процессе дано целое число. С помощью функций MPI\_Isend, MPI\_Irecv осуществить для всех процессов циклический сдвиг данных с шагом 2 и на кждом процессоре умножить на свой номер, переслав число из процесса 0 в процесс size-2 , из процесса size -2 в процесс size-4, …, из процесса 4 в процесс 2, из 2 в 0. В каждом процессе вывести полученное число.
12. Каждый процесс с помощью датчика случайных чисел задает переменную temp=(rank\* (rand(). На нулевом процессе с помощью датчика случайных чисел сформировать из 5 вещественных чисел: x[k]=k\*rand(), k=1,2,…5. С помощью функции MPI\_Isend, MPI\_Irecv разослать эти значения остальным процессам по цепочке 0→1, 1→2,…, size-2→size-1. Каждый процессор умножает полученные эти значения на переменную temp. Определить на каждом процессоре минимальный элемент.

Task 7

1. . В каждом процессе дано вещественное число. Используя функцию MPI\_Gather, переслать эти числа в главный процесс и вывести их в порядке возрастания рангов переславших их процессов (первым вывести число, данное в главном процессе).Найти максимальное число, и разослать всем процессорам.
2. . В главном процессе дан набор из K чисел, где K — количество процессов. Используя функцию MPI\_Scatter, переслать по одному числу в каждый процесс (включая главный) и вывести в каждом процессе полученное число. Наити максимальное число с помощью функций MPI\_REDUCE отправить в на процессе с номером 1.
3. . В главном процессе дан набор из 3K чисел, где K — количество процессов. Используя функцию MPI\_Scatter, переслать по 3 числа в каждый процесс (включая главный) и вывести в каждом процессе полученные числа и х суммы отправить , с помощью функций MPI\_REDUCE отправить на процессе с номером size-1.
4. MPI-процесс с номером 0 вводит с клавиатуры массив из 8 целых чисел. Затем с помощью функции MPI\_SCATTER рассылает по 4 процессам фрагменты этого массива. Каждый процесс печатает полученные данные. Произвести сложение всех элементов массива, распределенных по процессам, с помощью функций MPI\_REDUCE с получением результата на процессе с номером 2. Вычисленное значение распечатать
5. . В каждом процессе дан набор из 5 целых чисел. Используя функцию MPI\_Gather, переслать эти наборы в главный процесс и вывести их в порядке возрастания рангов переславших их процессов (первым вывести набор чисел, данный в главном процессе). Найти минимальное число и отправить всем остальным.
6. . В главном процессе дан набор из K чисел, где K — количество процессов. Используя функцию MPI\_Scatter, переслать по одному числу в каждый процесс (включая главный) и вывести в каждом процессе полученное число. Наити минимальное число с помощью функций MPI\_REDUCE отправить в на процессе с номером 1.
7. . В каждом процессе дано вещественное число. Используя функцию MPI\_Gather, переслать эти числа в главный процесс и вывести их в порядке возрастания рангов переславших их процессов (первым вывести число, данное в главном процессе).Найти максимальное число, и разослать всем процессорам.
8. . В главном процессе дан набор из K чисел, где K — количество процессов. Используя функцию MPI\_Scatter, переслать по одному числу в каждый процесс (включая главный) и вывести в каждом процессе полученное число. Наити максимальное число с помощью функций MPI\_REDUCE отправить в на процессе с номером 1.
9. . В главном процессе дан набор из 3K чисел, где K — количество процессов. Используя функцию MPI\_Scatter, переслать по 3 числа в каждый процесс (включая главный) и вывести в каждом процессе полученные числа и х суммы отправить , с помощью функций MPI\_REDUCE отправить на процессе с номером size-1.
10. MPI-процесс с номером 0 вводит с клавиатуры массив из 8 целых чисел. Затем с помощью функции MPI\_SCATTER рассылает по 4 процессам фрагменты этого массива. Каждый процесс печатает полученные данные. Произвести сложение всех элементов массива, распределенных по процессам, с помощью функций MPI\_REDUCE с получением результата на процессе с номером 2. Вычисленное значение распечатать
11. . В каждом процессе дан набор из 5 целых чисел. Используя функцию MPI\_Gather, переслать эти наборы в главный процесс и вывести их в порядке возрастания рангов переславших их процессов (первым вывести набор чисел, данный в главном процессе). Найти минимальное число и отправить всем остальным.
12. . В главном процессе дан набор из K чисел, где K — количество процессов. Используя функцию MPI\_Scatter, переслать по одному числу в каждый процесс (включая главный) и вывести в каждом процессе полученное число. Наити минимальное число с помощью функций MPI\_REDUCE отправить в на процессе с номером 1.