Определить сумму элементов матрицы с использованием функций MPI_Bcast, MPI_Scatterv, MPI_Reduce

Код

```
void Task1(int rowLength)
    {
        int procRank, procSize;

        MPI_Comm_size(MPI_COMM_WORLD, &procSize);
        MPI_Comm_rank(MPI_COMM_WORLD, &procRank);

        bool isMain = procRank == 0;

        double * matrix = NULL; // (rows, cols)
        int rowsCount = procSize;
        if (isMain)
        {
            matrix = new double[rowsCount * rowLength];
            for (int i = 0; i < rowsCount * rowLength; i++)
                matrix[i] = rand();
        }

        double * vector = new double[rowLength];
        MPI_Scatter(matrix, rowLength, MPI_DOUBLE,
                    vector, rowLength, MPI_DOUBLE, 0, MPI_COMM_WORLD);

        double vectorSumm = 0.0;
        for (int i = 0; i < rowLength; i++)
            vectorSumm += vector[i];

        cout << "matrix[" << procRank << "] summ: " << vectorSumm << endl;

        double matrixSumm = 0.0;
        MPI_Reduce(&vectorSumm, &matrixSumm, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

        if (isMain)
            cout << "matrix summ " << matrixSumm << endl;

        if (isMain)
            delete[] matrix;

        delete[] vector;
    }
```

Результаты работы

```
D:\My\Workspace\C++\MPICollectiveOperations\Debug>mpiexec.exe -n 4 "MPICollectiveOperations.exe"
matrix[0] summ: 288107
matrix[2] summ: 392756
matrix[1] summ: 293632
matrix[3] summ: 317309
matrix summ 1.2918e+06
```

Определить максимальный элемент массива и номер строки и столбца, в которых он находился с использованием функций MPI_Bcast, MPI_Scatterv, MPI_Reduce

Код

```
void Task2(int rowLength)
    {
        int procRank, procSize;

        MPI_Comm_size(MPI_COMM_WORLD, &procSize);
        MPI_Comm_rank(MPI_COMM_WORLD, &procRank);

        bool isMain = procRank == 0;

        int * matrix = NULL; // (rows, cols)
        int rowsCount = procSize;
        if (isMain)
        {
            matrix = new int[rowsCount * rowLength];
            for (int i = 0; i < rowsCount * rowLength; i++)
                matrix[i] = rand() % 100 + i;
        }

        int * vector = new int[rowLength];
        MPI_Scatter(matrix, rowLength, MPI_INT,
                    vector, rowLength, MPI_INT, 0, MPI_COMM_WORLD);

        int maxInVector[2] = { 0, procRank };
        for (int i = 0; i < rowLength; i++)
            if (vector[i] > maxInVector[0])
                maxInVector[0] = vector[i];

        cout << "Max in matrix[" << maxInVector[1] << "] summ: " <<
maxInVector[0] << endl;

        int maxInMatrix[2];
        MPI_Reduce(&maxInVector, &maxInMatrix, 1, MPI_2INT, MPI_MAXLOC, 0,
MPI_COMM_WORLD);

        if (isMain)
            cout << "Max in matrix found in row " << maxInMatrix[1] << " is " <<
maxInMatrix[0] << endl;

        if (isMain)
            delete[] matrix;

        delete[] vector;
    }
```

Результаты работы для 16 процессов и длины строки матрицы 20

```
D:\My\Workspace\C++\MPICollectiveOperations\Debug>mpiexec.exe -n 16 "MPICollectiveOperations.exe"
Max in matrix[0] summ: 111
Max in matrix[1] summ: 136
Max in matrix[2] summ: 137
Max in matrix[8] summ: 276
Max in matrix[4] summ: 179
Max in matrix[3] summ: 171
Max in matrix[9] summ: 284
Max in matrix[10] summ: 311
Max in matrix[12] summ: 343
Max in matrix[6] summ: 228
Max in matrix[5] summ: 215
Max in matrix[11] summ: 331
Max in matrix[13] summ: 368
Max in matrix[14] summ: 388
Max in matrix[7] summ: 248
Max in matrix[15] summ: 413
Max in matrix found in row 15 is 413
```

Определить вектор, получающийся в результате умножения матрицы на вектор с использованием функций MPI_Bcast, MPI_Scatterv, MPI_Gatherv

Код

```
void Task3(int rowLength)
    {
        int procRank, procSize;

        MPI_Comm_size(MPI_COMM_WORLD, &procSize);
        MPI_Comm_rank(MPI_COMM_WORLD, &procRank);

        bool isMain = procRank == 0;

        double * matrix = NULL; // (rows, cols)
        double * vector = new double[rowLength]; // (cols, 1)
        int rowsCount = procSize;
        if (isMain)
        {
            matrix = new double[rowsCount * rowLength];
            for (int i = 0; i < rowsCount * rowLength; i++)
                matrix[i] = rand() % 100 + 1;

            // vector = new double[rowLength];
            for (int i = 0; i < rowLength; i++)
                vector[i] = rand() % 100 + 1;
        }

        // Send vector b
        MPI_Bcast(vector, rowLength, MPI_DOUBLE, 0, MPI_COMM_WORLD);

        // Send matrix[procRank]
        double * matrixVector = new double[rowLength];
        MPI_Scatter(matrix, rowLength, MPI_DOUBLE,
                    matrixVector, rowLength, MPI_DOUBLE, 0, MPI_COMM_WORLD);

        // matrix[procRank] * vector => resultVector[procRank]
        double elementOfResultVector = 0.0;
        for (int i = 0; i < rowLength; i++)
            elementOfResultVector += matrixVector[i] * vector[i];

        // Send to master
        double * resultVector = NULL;
        if (isMain)
            resultVector = new double[procSize];

        MPI_Gather(&elementOfResultVector, 1, MPI_DOUBLE,
                    resultVector, 1, MPI_DOUBLE, 0, MPI_COMM_WORLD);

        // Print result vector to screen
        if (isMain)
        {
```

```
            cout << "Result vector is:" << endl;
            cout.precision(4);
            for (int i = 0; i < procSize; i++)
                cout << resultVector[i] << endl;
            cout << endl;
        }


        if (isMain)
        {
            delete[] matrix;
            delete[] vector;
            delete[] resultVector;
        }
        else
        {
            delete[] matrixVector;
        }
    }
```

Результаты работы для 16-и процессов

```
D:\My\Workspace\C++\MPICollectiveOperations\Debug>mpiexec.exe -n 16 "MPICollectiveOperations.exe"
Result vector is:
5676
2559
6015
7989
7071
2226
6996
8052
8589
3459
6819
2322
9933
2247
5112
4509
```