

Imię i nazwisko: _____

Sprawdzian z MPI

Zadanie 1

Program jest uruchamiany dla _____ procesów. Liczbą podawaną na standardowe wejście jest _____. Liczbą wypisywaną na standardowe wyjście jest: _____.

```
#include <iostream>
#include <boost/mpi.hpp>

namespace mpi = boost::mpi;

int main(int argc, char **argv)
{
    mpi::environment env(argc, argv);
    mpi::communicator world;

    if (world.rank())
        world.send(0, 1, world.size());
    else
    {
        int sum;

        std::cin >> sum;

        for (int i = 1; i < world.size(); ++i)
        {
            int value;
            world.recv(i, 1, value);
            sum += value;
        }

        std::cout << sum;
    }

    return 0;
}
```

Zadanie 2

Program jest uruchamiany dla _____ procesów. Liczbą podawaną na standardowe wejście jest _____. Liczbą wypisywaną na standardowe wyjście jest: _____.

```
#include <iostream>
#include <vector>
#include <boost/mpi.hpp>

namespace mpi = boost::mpi;

int main(int argc, char **argv)
{
    mpi::environment env(argc, argv);
    mpi::communicator world;

    int a, b, rnk;

    if (!world.rank())
    {
        std::cin >> a;
        rnk = (a + 2) % world.size();
    }

    broadcast(world, rnk, 0);

    a = rnk + world.rank();

    reduce(world, a, b, mpi::maximum<int>(), rnk);

    if (world.rank() == rnk)
        std::cout << b;

    return 0;
}
```

Zadanie 3

Program jest uruchamiany dla _____ procesów. Liczbą podawaną na standardowe wejście jest _____. Liczbą wypisywaną na standardowe wyjście jest: _____.

```
#include <iostream>
#include <vector>
#include <boost/mpi.hpp>

namespace mpi = boost::mpi;

int main(int argc, char **argv)
{
    mpi::environment env(argc, argv);
    mpi::communicator world;

    int a;
    int rank = world.rank();

    if (!rank)
        std::cin >> a;

    // Tworzy wektor z "size" elementami, każdy o wartości zmiennej "a".
    std::vector<int> tab(world.size(), a);

    int num;

    scatter(world, tab, num, 0);

    num += rank;

    gather(world, num, tab, 0);

    if (!rank)
        // Wypisz sumę pierwszego i ostatniego elementu tablicy "tab".
        std::cout << tab.front() + tab.back();

    return 0;
}
```

Zadanie 4

Program jest uruchamiany dla _____ procesów. Liczbą podawaną na standardowe wejście jest _____. Liczbą wypisywaną na standardowe wyjście jest: _____.

```
#include <iostream>
#include <vector>
#include <boost/mpi.hpp>

namespace mpi = boost::mpi;

class liczby
{
public:
    int a, b;

    template <class T>
    void serialize(T &t, const unsigned int)
    {
        t & a;
        t & b;
    }

    liczby() {}
    liczby(int a, int b) : a(a), b(b) {}
};

int main(int argc, char* argv[])
{
    mpi::environment env(argc, argv);
    mpi::communicator world;
    int rank = world.rank();
    int size = world.size();

    liczby t;

    if (!rank)
    {
        int a;
        std::cin >> a;
        t = liczby(a + size, a);
    }

    broadcast(world, t, 0);

    if (rank == 1)
        std::cout << t.a + t.b << std::endl;

    return 0;
}
```