#include <iostream>

#include <omp.h>

#include <iomanip>

using namespace std;

int main()

{

    // omp\_set\_num\_threads(4);

    double arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

    double max\_val = 0.0;

    double min\_val = 100.0;

    float avg = 0.0, sum = 0.0, sum\_val = 0.0;

    int i;

    #pragma omp parallel for reduction(min : min\_val)

    for (i = 0; i < 10; i++)

    {

        // printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(), i);

        if (arr[i] < min\_val)

        {

            min\_val = arr[i];

        }

    }

    cout<<"min\_val: "<<setprecision(5)<<min\_val<<endl<<endl;

    #pragma omp parallel for reduction(max : max\_val)

    for (i = 0; i < 10; i++)

    {

        // printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(), i);

        if (arr[i] > max\_val)

        {

            max\_val = arr[i];

        }

    }

    cout<<"max\_val: "<<setprecision(5)<< max\_val<<endl<<endl;

    #pragma omp parallel for reduction(+ : sum\_val)

    for (i = 0; i < 10; i++)

    {

        // printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(), i);

        sum\_val = sum\_val + arr[i];

    }

    cout<<"sum\_val: "<<setprecision(5)<< sum\_val<<endl<<endl;

#pragma omp parallel for reduction(+ : sum)

    for (i = 0; i < 10; i++)

    {

        // printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(), i);

        sum = sum + arr[i];

    }

    avg = sum / 10;

    cout<<"avg\_val: "<<setprecision(5)<< avg<<endl<<endl;

}