Implement Min, Max, Sum and Average operations using Parallel Reduction.

Name: Kaustubh Shrikant Kabra

Roll No: 37

BE COMPUTER SHIFT 1

Code:

```
#include <iostream>
#include <vector>
#include <omp.h>
#include <climits>
#include <random>


using namespace std;

void min_reduction(vector<int>& arr) {
  int min_value = INT_MAX;
  #pragma omp parallel for reduction(min: min_value)
  for (int i = 0; i < arr.size(); i++) {
    if (arr[i] < min_value) {
      min_value = arr[i];
    }
  }
  cout << "Minimum value: " << min_value << endl;
}

void max_reduction(vector<int>& arr) {
  int max_value = INT_MIN;
  #pragma omp parallel for reduction(max: max_value)
  for (int i = 0; i < arr.size(); i++) {
    if (arr[i] > max_value) {
      max_value = arr[i];
    }
  }
  cout << "Maximum value: " << max_value << endl;
}

void sum_reduction(vector<int>& arr) {
  int sum = 0;
   #pragma omp parallel for reduction(+: sum)
   for (int i = 0; i < arr.size(); i++) {
    sum += arr[i];
  }
  cout << "Sum: " << sum << endl;
}

void average_reduction(vector<int>& arr) {
  int sum = 0;
  #pragma omp parallel for reduction(+: sum)
  for (int i = 0; i < arr.size(); i++) {
    sum += arr[i];
  }
  cout << "Average: " << (double)sum / arr.size() << endl;
}

int main() {
    const int size = 100;
    vector<int> vec(size);
    random_device rd;
    mt19937 gen(rd());
    uniform_int_distribution<int> dis(1, 50);
    cout<<"Size of Vector :"<<vec.size()<<endl;

    cout<<"Vector values:";
    for (int i = 0; i < size; ++i) {
        vec[i] = dis(gen);
        cout<<vec[i]<<" ";
    }
    cout<<endl;

    double start_time = omp_get_wtime();
    min_reduction(vec);
    max_reduction(vec);
    sum_reduction(vec);
    average_reduction(vec);
    double end_time = omp_get_wtime();
    cout << "Runtime: " << end_time - start_time << " seconds" << endl;
}
```

Output:

Case 1: Random values between 1 to 50

```
PS C:\Practical_3> g++ -fopenmp .\parallel_reduction_openmp.cpp -o parallel_reduction

PS C:\Practical_3> .\parallel_reduction.exe

Size of Vector :100

Vector values:
```

```
29 37 18 46 1 30 44 26 48 49 9 34 16 27 36 40 13 4 29 17 17 30 12 2 9 47 49 1 33 39 24 17 4
18 1 21 43 46 15 33 47 29 43 1 29 15 45 37 1 36 14 23 17 12 47 44 24 36 49 45 39 17 36 34 9
31 25 4 33 39 34 42 43 8 27 1 16 25 17 5 21 35 29 32 35 38 29 24 16 39 50 38 30 49 20 9 21
23 8 42

Minimum value: 1

Maximum value: 50

Sum: 2681

Average: 26.81

Runtime: 0.00199986 seconds
```

Case 2: Random values between 1 to1000

```
PS C:\Practical_3> g++ -fopenmp .\parallel_reduction_openmp.cpp -o parallel_reduction

PS C:\Practical_3> .\parallel_reduction.exe

Size of Vector :100

Vector values:

562 727 349 916 7 595 873 516 960 976 170 662 317 529 702 789 256 75 575 330 339 584
239 31 173 929 967 20 654 780 478 337 67 343 14 415 842 908 283 652 921 571 849 8 561
287 890 730 2 703 263 453 338 224 935 873 479 718 979 886 768 337 703 672 167 606 498
67 647 769 665 827 852 145 534 3 307 494 329 98 414 693 563 626 697 755 577 462 320
778 995 752 583 977 398 178 410 450 143 836

Minimum value: 2

Maximum value: 995

Sum: 52706

Average: 527.06

Runtime: 0.00200009 seconds
```