Name:- Umakant Dodtalle
Roll No :- 4127
Practical No 3 :-Implement Min, Max, Sum and Average operations using Parallel Reduction

```
#include <iostream>
#include <vector>
#include <omp.h>
#include <time.h>

using namespace std;

int main() {
   const int size = 1000;
   vector<int> data(size);
   srand(time(0));
   for (int i = 0; i < size; ++i) {
      data[i] = rand() % 100;
   }

   int min_value = data[0];
#pragma omp parallel for reduction(min:min_value)
   for (int i = 0; i < size; ++i) {
      if (data[i] < min_value) {
         min_value = data[i];
      }
   }

   int max_value = data[0];
#pragma omp parallel for reduction(max:max_value)
   for (int i = 0; i < size; ++i) {
      if (data[i] > max_value) {
         max_value = data[i];
      }
   }

   int sum = 0;
#pragma omp parallel for reduction(+:sum)
   for (int i = 0; i < size; ++i) {
      sum += data[i];
   }

   float average = 0.0;
```

```
#pragma omp parallel for reduction(+:average)
    for (int i = 0; i < size; ++i) {
        average += static_cast<float>(data[i]) / size;
    }

    cout << "Minimum value: " << min_value << endl;
    cout << "Maximum value: " << max_value << endl;
    cout << "Sum of values: " << sum << endl;
    cout << "Average of values: " << average << endl;

    return 0;
}
```

Output :-

PS E:\HPC> cd "e:\HPC\" ; if ($?) { g++ HPC3.cpp -o HPC3 } ; if ($?) { .\HPC3 }
Minimum value: 0
Maximum value: 99
Sum of values: 48659
Average of values: 48.659