Name : Nikhil Khade

Roll No. : COBA14

Lab : HPC 3 (a)

Input :

```
#include <iostream>
#include <vector>
#include <omp.h>
using namespace std;

int parallelMin(vector<int> vec) {
    int min_val = vec[0];
    #pragma omp parallel for
    for (int i = 1; i < vec.size(); i++) {
        if (vec[i] < min_val) {
            min_val = vec[i];
        }
    }
    return min_val;
}

int parallelMax(vector<int> vec) {
    int max_val = vec[0];
    #pragma omp parallel for
    for (int i = 1; i < vec.size(); i++) {
        if (vec[i] > max_val) {
            max_val = vec[i];
        }
    }
    return max_val;
}

int parallelSum(vector<int> vec) {
```

```
    int sum = 0;
    #pragma omp parallel for
    for (int i = 0; i < vec.size(); i++) {
        sum += vec[i];
    }
    return sum;
}

float parallelAverage(vector<int> vec) {
    int sum = parallelSum(vec);
    float avg = float(sum) / vec.size();
    return avg;
}

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    vector<int> vec(n);
    cout << "Enter the elements: ";
    for (int i = 0; i < n; ++i) {
        cin >> vec[i];
    }

    int min_val = parallelMin(vec);
    cout << "Minimum value: " << min_val << endl;

    int max_val = parallelMax(vec);
    cout << "Maximum value: " << max_val << endl;

    int sum = parallelSum(vec);
    cout << "Sum of values: " << sum << endl;
```

```
    float avg = parallelAverage(vec);
    cout << "Average of values: " << avg << endl;

    return 0;
}
```

Output :