**Source Code:**

#include<iostream>

#include<omp.h> #include<bits/stdc++.h> using namespace std; void minimum(vector<int> array){ int min = INT\_MAX; double start = omp\_get\_wtime(); for(auto i = array.begin(); i != array.end();i++){ if(\*i < min){ min = \*i;

}

}

double end = omp\_get\_wtime(); cout << "Minimum Element: " << min << endl; cout << "Time Taken: " << (end-start) << endl;

int min\_ele = INT\_MAX; start = omp\_get\_wtime();

#pragma omp parallel for reduction(min: min\_ele) for(auto i = array.begin(); i != array.end();i++){ if(\*i < min\_ele){ min\_ele = \*i;

}

}

end = omp\_get\_wtime();

cout << "Minimum Element(Parallel Reduction): " << min\_ele << endl; cout << "Time Taken: " << (end-start) << endl;

}

void maximum(vector<int> array){ int max = INT\_MIN; double start = omp\_get\_wtime(); for(auto i = array.begin(); i != array.end();i++){ if(\*i > max){ max = \*i;

}

}

double end = omp\_get\_wtime(); cout << "Maximum Element: " << max << endl; cout << "Time Taken: " << (end-start) << endl;

int max\_ele = INT\_MIN; start = omp\_get\_wtime();

#pragma omp parallel for reduction(max: max\_ele) for(auto i = array.begin(); i != array.end();i++){ if(\*i > max\_ele){ max\_ele = \*i;

}

}

end = omp\_get\_wtime();

cout << "Maximum Element(Parallel Reduction): " << max\_ele << endl; cout << "Time Taken: " << (end-start) << endl;

}

void sum(vector<int> array){ int sum = 0;

double start = omp\_get\_wtime(); for(auto i = array.begin(); i != array.end();i++){

sum += \*i; }

double end = omp\_get\_wtime(); cout << "Summation: " << sum << endl; cout << "Time Taken: " << (end-start) << endl; sum = 0;

start = omp\_get\_wtime();

#pragma omp parallel for reduction(+: sum) for(auto i = array.begin(); i != array.end();i++){

sum += \*i; }

end = omp\_get\_wtime();

cout << "Summation(Parallel Reduction): " << sum << endl; cout << "Time Taken: " << (end-start) << endl;

}

void average(vector<int> array){ float avg = 0; double start = omp\_get\_wtime(); for(auto i = array.begin(); i != array.end();i++){ avg += \*i; }

double end = omp\_get\_wtime(); cout << "Average: " << avg / array.size() << endl; cout << "Time Taken: " << (end-start) << endl; avg = 0;

start = omp\_get\_wtime();

#pragma omp parallel for reduction(+: avg) for(auto i = array.begin(); i != array.end();i++){ avg += \*i; }

end = omp\_get\_wtime();

cout << "Average(Parallel Reduction): " << avg / array.size() << endl; cout << "Time Taken: " << (end-start) << endl;

} int main(){

cout << "Enter number of elements in array: "; int N; int MAX = 1000; cin >> N; vector<int> array; for(int i = 0; i < N; i++){

array.push\_back(rand() % MAX);

} minimum(array); maximum(array); sum(array); average(array); return 0;

}