Practical 3 :

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

using namespace std;

int main() {
   vector<int> data = {1, 5, 3, 7, 9, 2, 6, 4, 8};
   int min_val = INT_MAX;
   int max_val = INT_MIN;
   int sum = 0;

   #pragma omp parallel for reduction(min:min_val) reduction(max:max_val) reduction(+:sum)
   for (size_t i = 0; i < data.size(); ++i) {
      if (data[i] < min_val) {
         min_val = data[i];
      }
      if (data[i] > max_val) {
         max_val = data[i];
      }
      sum += data[i];
   }

   double average;
   #pragma omp parallel reduction(+:average)
   {
      average = (double)sum/ data.size();
   }

   cout << "Minimum: " << min_val << endl;
   cout << "Maximum: " << max_val << endl;
   cout << "Sum: " << sum << endl;
   cout << "Average: " << average << endl;

   return 0;
}
```

OUTPUT :

Minimum: 1

Maximum: 9

Sum: 45

Average: 5