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

using namespace std;

int minval(int arr[] , int n){
    int minval = arr[0];
    #pragma omp parallel for reduction(min : minval)
    for(int i = 0 ; i<n ; i++)
    {
        if(arr[i] < minval) {
            minval = arr[i];

        }
    }
    return minval;
}

int maxval(int arr[] , int n){
    int maxval = arr[0];
    #pragma omp parallel for reduction(max : maxval)
    for(int i = 0 ; i<n ; i++)
    {
        if(arr[i] > maxval) {
            maxval = arr[i];

        }
    }
    return maxval;
}


int sum1 (int arr[] , int n)
{
    int sum = 0;
    #pragma omp parallel for reduction(+: sum)
     for(int i = 0 ; i<n ; i++)
    {
        sum += arr[i];

    }
    return sum;
}

double average(int arr[] , int n)
{
   return (double)sum1(arr,n) /n;
}

int main()
{
   int n = 5;
   int arr[] = {1,2,3,4,5};
   cout<<"minimum value: " <<minval(arr,n) << endl;
   cout<<"maximum value: " <<maxval(arr,n) << endl;
   cout<<"sum is: " <<  sum1(arr,n) << endl;
   cout<<"Average is: "<<average(arr,n) << endl;

   return 0;
}
```