Code:

```
#include <iostream>
#include <omp.h>
#include <time.h>
#include <iomanip>
#include <limits>
using namespace std;

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

int min(int a[],int n) {
  int v = a[0];
  #pragma omp parallel for reduction(min:v)
  for (int i = 0; i < n; i++) {
    if(a[i] < v)
      v = a[i];
  }
  return v;
}

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

float avg(int a[],int n) {
  return sum(a,n)/n;
}

int main()
{
  int a[100];
  int cnt=0;
  for(int i = 0;i<100;i++)
    { a[i] = i+5;
     cnt = cnt +1;}
  cout<<" Input data is :" ;
  for(int i = 0;i<cnt;i++)
    cout<<" " <<a[i];
  cout<<"\n";



     struct timespec start, end;

    // start timer.
    // clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &start);
    //lo clock_gettime(CLOCK_REALTIME, &start);
    clock_gettime(CLOCK_MONOTONIC, &start);

    // unsync the I/O of C and C++.
    ios_base::sync_with_stdio(false);

  cout<<"Sum : "<<sum(a,100);



    // stop timer.
    // clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &end);
    // clock_gettime(CLOCK_REALTIME, &end);
    clock_gettime(CLOCK_MONOTONIC, &end);

    // Calculating total time taken by the program.
    double time_taken;
    time_taken = (end.tv_sec - start.tv_sec) * 1e9;
    time_taken = (time_taken + (end.tv_nsec - start.tv_nsec)) * 1e-9;

    cout << "Time taken by program is : " << fixed
       << time_taken << setprecision(5);
    cout << " sec" << endl;

    cout<<"\n";



  cout<<" Sum of 100 numbers: "<<sum(a,100);
  cout<<"\n";
  cout<<" Minimum number from 100 numbers: "<<min(a,100);
  cout<<"\n";
  cout<<" Maximum number from 100 numbers: "<<max(a,100);
  cout<<"\n";
```

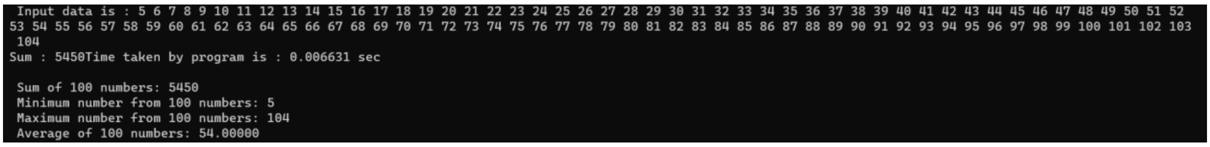
```
  cout<<" Average of 100 numbers: "<<avg(a,100);


  cout<<"\n";

  return 0;

}
```

Output:



```
 Input data is : 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103
 104
Sum : 5450Time taken by program is : 0.006631 sec

 Sum of 100 numbers: 5450
 Minimum number from 100 numbers: 5
 Maximum number from 100 numbers: 104
 Average of 100 numbers: 54.00000
```