#include <iostream>

#include <vector>

#include <algorithm>

#include <numeric>

#include <limits>

// #include <omp.h>

using namespace std;

int findMin(vector<int>& data) {

int minVal = numeric\_limits<int>::max();

for (int i = 0; i < data.size(); i++) if (data[i] < minVal) minVal = data[i];

return minVal;

}

int findMax(vector<int>& data) {

int maxVal = numeric\_limits<int>::min();

for (int i = 0; i < data.size(); i++) if (data[i] > maxVal) maxVal = data[i];

return maxVal;

}

int findSum(vector<int>& data) {

int sum = 0;

for (int i = 0; i < data.size(); i++) sum += data[i];

return sum;

}

double findAverage(vector<int>& data) {

double sum = 0;

for (int i = 0; i < data.size(); i++) sum += data[i];

return sum / data.size();

}

int main() {

vector<int> data(100);

generate(data.begin(), data.end(), [](){ return rand() % 100; });

cout << "Data :" << endl;

for (int i = 1; i < data.size()+1; i++) {

cout << " " << data[i] ; if(i%10==0) cout << endl;

} cout << endl;

cout << "Minimum : " << findMin(data) << endl;

cout << "Maximum : " << findMax(data) << endl;

cout << "Sum : " << findSum(data) << endl;

cout << "Average : " << findAverage(data) << endl;

return 0;

}

// • Output:

// Data :

// 86 77 15 93 35 86 92 49 21 62

// 27 90 59 63 26 40 26 72 36 11

// 68 67 29 82 30 62 23 67 35 29

// 2 22 58 69 67 93 56 11 42 29

// 73 21 19 84 37 98 24 15 70 13

// 26 91 80 56 73 62 70 96 81 5

// 25 84 27 36 5 46 29 13 57 24

// 95 82 45 14 67 34 64 43 50 87

// 8 76 78 88 84 3 51 54 99 32

// 60 76 68 39 12 26 86 94 39 0

// Minimum : 2

// Maximum : 99

// Sum : 5184

// Average : 51.84