TP 4 - Problem 3 - I

```cpp
#include <algorithm>
#include <iostream>
#include <numeric>
#include <stdexcept>
#include <vector>

#define SHOW(arg) std::cout << "Macro SHOW “" #arg "”: " << (arg) << '\n';

double mean(std::vector<double> const &v) {
  if (v.empty())
    throw std::domain_error{"Attempt to compute the mean of the elements "
                            "from an empty vector."};
  // Compute the mean using the accumulate algorithm of the STL.
  return std::accumulate(v.begin(), v.end(), 0.) / v.size();
  // double sum{v[0]};
  // for (size_t i{1}; i < v.size(); ++i)
  //   sum += v[i];
  // return sum / v.size();
}
```

TP 4 - Problem 3 - II

```
20  double standardDeviation(std::vector<double> const &v) {
21    if (v.empty())
22      throw std::domain_error{
23          "Attempt to compute the standart deviation of the elements "
24          "from an empty vector."};
25    // [sum_{i=1}^n (x_i - xb)^2] / (n-1) =
26    // (sum_{i=1}^n x_i^2 - n xb^2) / (n-1)
27    if (v.size() == 1)
28      return 0;
29    // Compute the standard deviation using the accumulate and inner_product
30    // algorithms of the STL.
31    double const s1{std::accumulate(v.begin(), v.end(), 0.)};
32    double const s2{std::inner_product(v.begin(), v.end(), v.begin(), 0.)};
33    return (s2 - s1 * s1 / v.size()) / (v.size() - 1);
34    // double s1{v[0]};
35    // double s2{v[0]};
36    // for (size_t i{1}; i < v.size(); ++i)
37    //   s1 += v[i], s2 += v[i]*v[i];
38    // return (s2 - s1 * s1 / v.size()) / (v.size() - 1);
39  }
```

TP 4 - Problem 3 - III

```
40  double min(std::vector<double> const &v) {
41    if (v.empty())
42      throw std::domain_error{"Attempt to compute the min of the elements "
43                              "from an empty vector."};
44    // Compute the min using the min algorithm of the STL (ranges library).
45    return std::ranges::min(v);
46    // double min{v[0]} ;
47    // for (size_t i{1}; i < v.size(); ++i)
48    //   if (v[i] < min)
49    //     min = v[i];
50    // return min;
51  }
52  double max(std::vector<double> const &v) {
53    if (v.empty())
54      throw std::domain_error{"Attempt to compute the min of the elements "
55                              "from an empty vector."};
56    // Compute the max using the max algorithm of the STL (ranges library).
57    return std::ranges::max(v);
58    // double max{v[0]} ;
59    // for (size_t i{1}; i < v.size(); ++i)
```

TP 4 - Problem 3 - IV

```
60      //   if (v[i] > max)
61      //     max = v[i];
62      // return max;
63    }
64
65    int main() {
66      SHOW(mean(std::vector<double>{1, 2, 3}))
67      SHOW(standardDeviation(std::vector<double>{1}))
68      SHOW(standardDeviation(std::vector<double>{1, 2}))
69      SHOW(standardDeviation(std::vector<double>{1, 2, 3}))
70      SHOW(min(std::vector<double>{1, 2, 3}))
71      SHOW(max(std::vector<double>{1, 2, 3}))
72      return 0;
73    }
```

TP 4 - Problem 3 - V

Output:

```
1  Macro SHOW “mean(std::vector<double>{1, 2, 3})”: 2
2  Macro SHOW “standardDeviation(std::vector<double>{1})”: 0
3  Macro SHOW “standardDeviation(std::vector<double>{1, 2})”: 0.5
4  Macro SHOW “standardDeviation(std::vector<double>{1, 2, 3})”: 1
5  Macro SHOW “min(std::vector<double>{1, 2, 3})”: 1
6  Macro SHOW “max(std::vector<double>{1, 2, 3})”: 3
```