TP 4 - Problem 5 - I

```cpp
#include <cassert>
#include <cmath>
#include <iostream>
#include <numbers>

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

// Abstract base class, i.e. the specification of a interface.
struct RealFunction {
  virtual ~RealFunction(){};
  virtual double evaluate(double x) const = 0;
};

// Free function to compute an definite integral by trapezoid
// approximation:
//         h/2 [f(x_0) + 2*f(x_1) + ... + 2*f(x_N-1) + f(x_N)]
// with h = (b-a)/nPoints and x_n = a + n h, n = 0, ..., N.
double integral(RealFunction const &f, double a, double b, int nPoints) {
  assert(nPoints >= 3);
```

TP 4 - Problem 5 - II

```
20    double df{f.evaluate(a)};
21    for (int i{1}; i < nPoints; ++i)
22      // Do not use the “a + n h” expression as errors are cumulative.
23      df += 2 * f.evaluate(((1 - i) * a + i * b) / nPoints);
24    df += f.evaluate(b);
25    return (b - a) / nPoints / 2 * df;
26  }
27
28  // Definition of the sin function.
29  struct SinFunction : public RealFunction {
30    double evaluate(double x) const { return std::sin(x); };
31    ~SinFunction(){};
32  };
33
34  int main() {
35    SHOW(integral(SinFunction{}, 0, 2 * std::numbers::pi, 3))
36    SHOW(SinFunction{}.evaluate(0))
37    SHOW(SinFunction{}.evaluate(std::numbers::pi))
38    SHOW(SinFunction{}.evaluate(2 * std::numbers::pi))
```

TP 4 - Problem 5 - III

```
39      return 0;
40  }
```

Output:

```
1  Macro SHOW “integral(SinFunction{}, 0, 2 * std::numbers::pi, 3)”:
    ↪  4.41084e-16
2  Macro SHOW “SinFunction{}.evaluate(0)”: 0
3  Macro SHOW “SinFunction{}.evaluate(std::numbers::pi)”: 1.22465e-16
4  Macro SHOW “SinFunction{}.evaluate(2 * std::numbers::pi)”: -2.44929e-16
```