

```

#include <iostream>

struct Number {
    int x;
};

template<typename T>
T add(const T a, const T b) {
    return a + b;
}

template<> // specialisation
Number add(const Number a, const Number b) {
    return Number{a.x + b.x};
}

template<typename T, typename F>
T doSomething(T a, T b, F func) {
    return func(a, b);
}

template<typename T>
struct Fonct { // foncteur
    T operator()(const T a, const T b) { return a + b; }
};

int main() {

    std::cout << add(5, 2) << std::endl;
    std::cout << add(Number{5}, Number{2}).x << std::endl;

    std::cout << doSomething(1, 2, [](const int a, const int b) { return a + b; }) << std::endl;
    std::cout << doSomething(1, 2, Fonct<int>{}) << std::endl;

    // affiche
    // 7
    // 7
    // 3
    // 3

    return 0;
}

```