TP 2 - Problem 1 - I

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

// Returns the argument.
int f(int i) { return i; }
// Returns the square of the argument.
int g(int i) { return i * i; }

// Declaration mirrors usage.
int sum(int (*fun_ptr)(int), int n) {
  assert(n > 0);
  int s{};
  for (int i{}; i < n; ++i)
    // Call operator () is a higher priority than dereferenciation
    // operator *: extra parenthesis are mandatory.
    s += (*fun_ptr)(i);
  return s;
}

```

TP 2 - Problem 1 - II

```
20  // The typedef names are aliases for existing types and are not
21  // declarations of new types.
22  typedef int (*FunIntRetIntPtr)(int);
23  // Typedef name simplifies syntax.
24  int sum_(FunIntRetIntPtr fun_ptr, int n) {
25    assert(n > 0);
26    int s{};
27    for (int i{}; i < n; ++i)
28      s += (*fun_ptr)(i);
29    return s;
30  }
31
32  int main() {
33    std::cout << sum(f, 10) << '\n';
34    std::cout << sum(g, 10) << '\n';
35    std::cout << sum_(f, 10) << '\n';
36    std::cout << sum_(g, 10) << '\n';
37    return 0;
38  }
```

TP 2 - Problem 1 - III

Output:

```
45
285
45
285
```