

W rozwiązaniach można używać dowolnych udogodnień z biblioteki standardowej C++ oraz bibliotek Boost.

Singleton Logger jest używany w aplikacji współbieżnej, pokazanej niżej. Jeżeli trzeba, to popraw kod w ramce.

```
class Logger {
public:
    static Logger* getInstance() {
        if(!instance)
            instance = new Logger();
        return instance;
    }

    void msg(const string& s) {
        cout << ++counter_ << ":" << s << endl;
    }
private:
    Logger() : counter_(0) {}
    Logger(const Logger&) = delete;

    int counter_;
    static Logger* instance;
};

Logger* Logger::instance = nullptr;
```

Zadanie 2 (1pkt)

Liczba liter Twojego nazwiska to stała NAME_SIZE.

```
const int NAME_SIZE = ;
```

```
using V = variant<int, string, double>;
int count(const std::vector<V>& vec) {
    struct Visitor
    {
        Visitor() : counter_(0) {}
        void operator()(const int& i) { counter_ += i; }
        void operator()(const string& s) { counter_ += s.size(); }
        void operator()(const double& d) { counter += static_cast<int>(d); }
        int counter_;
    } visitor;
    for_each(vec.begin(), vec.end(), [&](const V& v){ std::visit(visitor, v); });
    return visitor.counter_;
};

vector<V> vec;
vec.push_back(NAME_SIZE); vec.push_back(NAME_SIZE-3); vec.push_back(string("ZPR25L"));
```

Podaj wartość zwracaną przez funkcję count (vec)

Zadanie 3 (2pkt)

Napisz szablon funkcji, która zwraca medianę (wartość środkową) 3 wartości danego typu. Zakładamy, że typ dostarcza operator<, co jest badane przez mechanizm konceptów (C++20). Proszę założyć, że to badanie jest poprawne.

```
template<typename T>
concept LessThenComparable = requires(T a, T b)
{
    { a < b } -> std::convertible_to < bool >;
};
```

```
template<typename T>
requires LessThenComparable<T>
T median(T a, T b, T c) {
```

Pytanie 1 (1pkt)

Jakie treści proponujesz dodać do ZPR w przyszłych edycjach?

Ile godzin poświęciłeś na ZPR (uwzględnij wszystko, tzn. wykład, kolokwia, projekt i inne)?

Uwagi do prowadzącego (R.Nowak):

[illegible]**Zadanie 4 (2pkt)**

Wymień zalety dedukcji typów w C++. Podaj przykłady i uzasadnij.

Zadanie 5 (2pkt)

Zaproponuj implementację funkcji, która liczy obwód dowolnej z poniższych figur.

```
struct Circle { double diameter; };  
struct Rectangle { double sides[2]; };  
struct Triangle { double lengths[3]; };
```

Zwróć uwagę, że nie każda kombinacja długości boków daje prawidłową figurę. Warunki dodatkowe: rozwiązaniem ma być pojedyncza funkcja, która nie może rzucać wyjątków.

Uwagi do prowadzącego (W. Wysota):