Imię i nazwisko: **22L kol2**

Proszę wpisywać odpowiedzi w miejscach na to przeznaczonych

Zadanie 1 (2pkt)

Proszę odpowiedzieć na poniższe pytania:

- Czy (i ewentualnie jak) można mierzyć "utrzymywalność" kodu?

- Czy analiza dynamiczna ma przewagę nad statyczną?

Zadanie 2 (2pkt)

Dana jest funkcja:

```
std::shared_ptr<Item> Service::method(const Other* obj, int x) {
    if (obj != nullptr && obj->hasSomeProperty())
        return nullptr;
    if (x < -10)
        throw InvalidArgument(x);
    return x > 5 ? Item::Default : std::make_shared<Item>(x);
}
```

Ile przypadków testowych należy napisać, by uzyskać:

- 100% pokrycia linii kodu,
- 100% pokrycia gałęzi kodu (mierzonych narzędziem w rodzaju LCOV).

Podaj wartości argumentów konieczne do uzyskania wspomnianych pokryć.

Zadanie 3 (2pkt)

Co jest nie tak z poniższym kodem?

```
int Service::process(const Request& r, bool ignored) {
    // sprawdzamy, czy usługa działa
    if (isConfigured() && registry->isEnabled(this) && subservice != nullptr && subservice->isConfigured()
                    && subservice->isActive() && registry->isEnabled(subservice))
    {
        if (!r.header.topic.items.contain(Request::Header::Type::Active))
            return -2;
        auto tmp = r.body.items[3].data;
        if (tmp > 5) {
            tmp = 15 * itemCount();
            if (ignored)
                throw IncorrectRequest(r);
            tmp = sendResponse(r, tmp);
            if (tmp != 0)
                return tmp;
            else
                return 0;
        } else {
            if (ignored)
                throw IncorrectRequest(r);
            tmp = sendResponse(r, 15 * itemCount() - 2);
            if (tmp != 0)
               return tmp;
            return 0;
        }
    }
    else
    {
        return -4;
    }
    return -5;
}
```

Proszę opisać jak należałoby poprawić ten kod (jakie operacje i w jaki sposób wykonane, kod nie jest konieczny).

Uwagi do prowadzącego (K. Grochowskiego):

Imię i nazwisko: **22L kol2**

Proszę wpisywać odpowiedzi w miejscach na to przeznaczonych

Zadanie 4 - współbieżne wzorce projektowe (2pkt)

Program oblicza ilość liczb pierwszych, wykorzystując wiele wątków. W aplikacji występuje wyścig. Proszę to eyeliminować dbając o skalowalność.

W funkcji 'main' występują obiekty 'ref(x)'. Są to obiekty pomocnicze, które zapobiegają tworzeniu kopii obiektu, więc np. 'thread thrd1(ref(p1));' oznacza, że funkcja wątku obliczeniowego to będzie 'operator()' obiektu 'p1' (a nie jego kopii). Nie jest to kluczowe do znalezienia rozwiązania.

```
//bada, czy n jest liczba pierwsza (tej funkcji nie zmieniamy)
bool isPrime(long n) {
   if( n < 2 ) return false; // '0' and '1' is not prime
   if( n < 4 ) return true; // '2' and '3' is prime
   if( n % 2 == 0) return false;
   for(long i = 3; i * i <= n; i = i + 2)
       if(n % i == 0) return false;
   return true;
}
```

```
class ThreadPrime {
public:
    ThreadPrime(int n, atomic<int>& num) : n_max(n), num_primes(num) { }
    void operator()() {
        for(int i = 1; i < n_max; ++i ) {
            if(isPrime(i) )
                ++num_primes;
        }
    }
private:
    int n_max;
    atomic<int>& num_primes;
};


int main () {
    cout << "Program␣podaje␣ilosc␣liczb␣pierwszych␣mniejszych␣niz␣N" << endl;
    cout << "podaj␣N:";
    int n_max = 0;
    cin >> n_max;

    atomic<int> num_primes = 0;

    ThreadPrime t1(n_max, num_primes);
    ThreadPrime t2(n_max, num_primes);
    ThreadPrime t3(n_max, num_primes);

    thread thrd1( ref(t1) );
    thread thrd2( ref(t2) );
    thread thrd3( ref(t3) );
    thrd1.join();
    thrd2.join();
    thrd3.join();
    cout << "num␣primes:" << num_primes << endl;
    cout << endl;
    return 0;
}
```

Uwagi do prowadzącego (R. Nowaka):

Zadanie 5 - współbieżne wzorce projektowe (2pkt)

Kolekcja Out nie jest używana zanim nie zostanie uzupełniona. W przedstawionym rozwiązaniu wykorzystywane są synchroniczne blokujące operacje wysyłania. W aplikacji występuje wyścig. Proszę to eyeliminować dbając o skalowalność.

```
class Data {}; //nie zamieszczono skladowych
using PData = shared_ptr<Data>;
using Event = function<void(void)>;
class Socket { //sluzy do komunikacji, implementacja nieistotna
public:
    Socket(boost::asio::io_service& io);
    void send(PData d); //wysylanie synchroniczne blokujace
    void async_send(PData d, Event event); //wys. asynchroniczne
    //inne metody sa nieistotne.
};
```

W zadaniu są sprytne wskaźniki, o nazwie 'PData'. Ich znajomość nie jest konieczna, aby znaleźć rozwiązanie.
W zadaniu są kontenery na funkcję 'function<void(void)>'. Ich znajomość nie jest konieczna, aby znaleźć rozwiązanie.
W funkcji 'main' występują obiekty 'ref(x)'. Ich opis jest przy poprzednim zadaniu. Nie jest to kluczowe do znalezienia rozwiązania.

```
struct Out {
    std::vector<PData> v;
    mutex m;
    //pobiera ostatni element z kolekcji i go usuwa.
    PData pop() {
        PData d; //init on nullptr
        if( ! v.empty() ) {
            d = v.back();
            v.pop_back();
        }
        return d;
    }
};

class Thread {
public:
    Thread(boost::asio::io_service& i, Out& o) : io(i), out(o) { }
    void operator()() {
        Socket s(io); //local socket
        PData d; //init on nullptr
        do {
            d = out.pop();
            if(d != nullptr) {
                s.send(d);
            }
        } while( d != nullptr );
    }
private:
    boost::asio::io_service& io;
    Out& out;
};

int main() {
    Out out;
    for(int i=0;i<500;++i)
        out.v.push_back( PData(new Data()) ); //tworzenie paczki danych, tego nie poprawiamy

    boost::asio::io_service io;

    Thread t1(io, out);
    Thread t2(io, out);
    Thread t3(io, out);

    thread thrd1( ref(t1) );
    thread thrd2( ref(t2) );
    thread thrd3( ref(t3) );
    thrd1.join();
    thrd2.join();
    thrd3.join();
    return 0;
}
```

Część druga (0.5 pkt) - napisz, które mechanizmy można wyelimionować w tym problemie: (a) wiele wątków, (b) sekcje krytyczne, (c) pętlę zdarzeń, jeżeli zdecydujemy się wykorzystać asynchroniczną obsługa urządzeń, tutaj wołanie metody 'async_send' dla Socket?

a[ ], b[ ], c[ ]