

Przyjąć, że udostępniona jest przestrzeń nazw std

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

Zadanie 1 (2pkt)

Przedstawiona aplikacja zachowuje się niepoprawnie. Popraw funkcję wykonywaną w wątkach użytkownika.

```
static const int NUM = 100;
struct Queue {
    queue<int> data_;
    mutex mutex_;
};
;
int main() {
    Queue q1, q2, q3;
    q1.data_.push_back(100);
    Producer p1(1, q1, q2);
    Producer p2(2, q2, q3);
    Producer p3(3, q3, q1);

    thread thrd1( ref(p1) ); thread thrd2( ref(p2) ); thread thrd3( ref(p3) );

    thrd1.join(); thrd2.join(); thrd3.join();
    return 0;
}
```

```
struct Producer {

    Producer(int id, Queue& in, Queue& out) : id_(id), in_(in), out_(out) {}

    void operator() () {
        for(int num = 0; num < NUM; ) {
            lock_guard<mutex> lock_in(in_.mutex_);
            lock_guard<mutex> lock_out(out_.mutex_);
            if(!in_.data_.empty() ) {
                int x = in_.data_.front();
                in_.data_.pop_front();

                //tutaj przetwarza dane lokalnie

                out_.data_.push_back(x+1);
                ++num;
            }
        }
    }
    int id_;
    Queue& in_;
    Queue& out_;
};
```

Pytanie 1 (0.5pkt)

Jakie treści proponujesz dodać do ZPR w przyszłych edycjach?

Pytanie 2 (0.5pkt)

Ile godzin poświęciłeś na wykonanie projektu z ZPR?

Ile godzin poświęciłeś na pozostałe elementy przedmiotu (wykłady, kolokwia, zadanie dodatkowe, itd.)?

Zadanie 2 (1pkt)

Podaj wydruk. Liczba liter Twojego nazwiska (stała NAME_SIZE jest użyta w zadaniu):

```
const int NAME_SIZE = ;
```

Dokumentacja związana z funkcjami użytymi w zadaniu.

- Computes the absolute value of an integer number.

```
int std::abs(int);
```

- Finds the smallest element in the range [first, last). Elements are compared using operator<.

```
template<class ForwardIt > ForwardIt std::min_element( ForwardIt first, ForwardIt last );
```

- Finds the smallest element in the range [first, last). Elements are compared using the given binary comparison function comp

```
template< class ForwardIt, class Compare >  
ForwardIt std::min_element( ForwardIt first, ForwardIt last, Compare comp );
```

```
int main() {  
    vector<int> v = {10, 0, 5, 15, 3};  
    cout << *min_element(v.begin(), v.end())  
        << *min_element(v.begin(), v.end(), [](int a, int b){ return abs(NAME_SIZE-a) < std::abs(NAME_SIZE-b);})  
        << *min_element(v.begin(), v.end(), [](int a, int b){ return abs(2-a) < std::abs(2-b);})  
        << endl;  
    return 0;  
}
```

Zadanie 3 (2pkt)

Proszę dostarczyć szablon fast_copy, który kopiuje tablice. Użyownik tego szablonu gwarantuje, że obszary pamięci się nie pokrywają. Proszę używać memcpy jeżeli jest to poprawne. Dokumentacja związana z zadaniem:

- `template< class T > struct std::is_trivially_copyable;`

Objects of trivially-copyable types that are the only C++ objects that may be safely copied with `std::memcpy`. If T is a trivially copyable type, `std::is_trivially_copyable<T>::value` (constant member) is equal to **true**.

For any other type, `std::is_trivially_copyable<T>::value` is **false**.

- Copies count bytes from the object pointed to by src to the object pointed to by dest. `std::memcpy` is meant to be the fastest library routine for memory-to-memory copy.

```
void* std::memcpy(void* destination, const void* source, size_t count)
```

```
template<class T> void fast_copy(const T* first, const T* last, T* result) {
```

Uwagi do prowadzącego (R.Nowak):

Imię i nazwisko:

F

czas: 60 min

Przyjąć, że udostępniona jest przestrzeń nazw std

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

Przyjąć, że udostępniona jest przestrzeń nazw std

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

Zadanie 4 (2pkt)

Dane są klasy Point, Rectangle, Circle. Figurą jest koło i prostokąt.

- zaproponuj typ Shape,
- zaimplementuj funkcję leftEdge (deklaracja jest podana niżej), która pobiera wektor zawierający kolekcję figur i zwraca wartość, która reprezentuje współrzędną x skrajnie lewego punktu zajmowanego przez ten zestaw figur.

```
struct Point { double x, y; };  
struct Rectangle { Point topLeft, bottomRight; };  
struct Circle { Point centre; double radius; };
```

```
using Shape =  
  
auto leftEdge(const std::vector<Shape> & shapes) {
```

Uwagi do prowadzącego (W. Wysota):

Zadanie 5 (2pkt)

Zaimplementuj algorytm, który pobiera zakres niepowtarzalnych elementów oraz wartość. Algorytm zwraca liczbę elementów, które są większe od mediany (wartości środkowej) zakresu i jednocześnie mniejszych niż podana wartość.

Algorytm ma działać dla dowolnych typów, które obsługują wymagane operacje i ma być optymalny wydajnościowo.

Można założyć, że otrzymany zakres pozwala na wielokrotną iterację po elementach oraz swobodne skakanie z dowolnego elementu na dowolny inny (innymi słowy modeluje `random_access_range`). Dozwolona jest modyfikacja kolejności elementów otrzymanego zakresu.

```
auto process(auto from, auto to, auto value) {
```