Imię i nazwisko: | | | | | | | | | | | | | | | | | | | | | | | | | | **24Z-2** 60 min

Przyjąć, że udostępniona jest przestrzeń nazw std i boost

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

Zadanie 1 (3pkt)

Popraw wydajność funkcji`countProper` na platformach współbieżnych. Funkcja ta zlicza obiekty `Data`, które mają odpowiednie właściwości. Obiekty typu `Data` są duże, funkcja `isProper` liczy się długo.

```
using Data;
bool isProper(const Data& d) {
  //funkcja działa długo, nie używa innych obiektów niż argument
}
```

```
struct InData {
   InData(const vector<Data>& r) : records_(r), counter_(0) {}
   const vector<Data>& records_;
   mutex m_;
   int counter_;
};

struct Thread {
   Thread(InData& d) : data_(d), currentIdx_(0) {}

   void setIdx(int i) { currentIdx_ = i; }
   void operator()() {
      std::lock_guard<std::mutex> lock( data_.m_ );
      if(isProper(data_.records_[currentIdx_]) ) {
         data_.counter_ += 1;
      }
   }
private:
   int currentIdx_;
   InData& data_;
};

int countProper(const vector<Data>& records) {
   InData input(records);

   const int NUM_THREADS = 3;
   std::vector<Thread> vec_thread(NUM_THREADS, Thread(input) );

   for(int i = 0, epoch_size = 0; i < records.size(); i += epoch_size) {
      int still_not_used = records.size() - i;
      epoch_size = std::min(NUM_THREADS, still_not_used); //w tym obiegu tyle bedzie przetwarzac

      vector<thread> thrds;
      for(int j = 0; j < epoch_size; ++j) {
         vec_thread[j].setIdx(i+j);
         thrds.push_back( thread(ref(vec_thread[j]) ) );
      }
      for(int j = 0; j < epoch_size; ++j) {
         thrds[j].join();
      }
   }
   return input.counter_;
}
```

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(string("ZPR24Z")); vec.push_back(1.1);
```

Podaj wartość zwracaną przez funkcję `count(vec)` [ ]

Zadanie 3 (1pkt)

Popraw funkcję `readInt()`, aby obsługiwała wszystkie liczby kodowane przez int. Obecna wersja używa niektórych wartości do sygnalizowania błędu.

Strumień istream jest skonfigurowany aby nie zgłaszać wyjątków, `operator>>(int& value)` dokonuje poprawnej konwersji, tzn. pomija białe znaki, wczytuje cyfry i dostarcza liczbę całkowitą, ustawia flagi, które można badać, np. `failbit()` jest ustawiany, gdy błąd formatowania, np. litery zamiast cyfr.

Do sygnalizacji błędu wykorzystaj obiekty klas `ErrorEmptyInput`, `ErrorBadInput`

```
class ErrorEmptyInput : public std::exception { };
class ErrorBadInput : public std::exception { };
```

```
static const int ERROR_BAD = 0;
static const int ERROR_EMPTY = -1;

int readInt(istream& is) {
   if(is.peek() == EOF) {
      return ERROR_EMPTY;
   }
   int out = 0;
   is >> out;
   if(is.fail() || is.bad() )
      return ERROR_BAD;
   else
      return out;
}
```

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):

Zadanie 4 (2pkt)

W koszyku piknikowym (kolekcja elementów) znajdują się owoce i napoje butelkowane. Każdy element koszyka ma jakąś cenę, a napoje butelkowane oprócz tego mają jeszcze kaucję. Zaproponuj i zaimplementuj sposób reprezentacji zawartości koszyka oraz funkcję `totalValue`, która pobierze zawartość koszyka i zwróci sumaryczną wartość tego koszyka z uwzględnieniem kaucji za butelki.

Uwaga! W rozwiązaniu nie można użyć pętli `for`, `while` ani `std::for_each`.

```
struct Fruit {
  int value;
};
struct Drink {
   int value;
   int deposit;
};
```

```
int totalValue(
```

Zadanie 5 (2pkt)

Napisz szablon funkcji, która na wejściu otrzyma kolekcję elementów i zwróci wartość środkowego elementu (medianę).

Uwagi:

- można założyć, że kolekcja zawiera przynajmniej jeden element;
- można założyć, że kolekcja posiada funkcje begin() i end() zwracające iteratory;
- w rozwiązaniu nie można użyć pętli ani std::for_each;
- funkcja może zmodyfikować kolejność elementów w kolekcji jako efekt uboczny swojego działania;
- funkcja ma działać dla różnych typów danych, tzn. możemy do niej podać zarówno np. wektor intów jak i ciąg znaków;
- złożoność rozwiązania ma być amortyzowana liniowa (std::sort się nie kwalifikuje)

Przykład wołania

```
int main() {
   auto vec = std::vector{6, 6, 6, 1, 1, 1, 4};
   std::println("{}", median(vec));
   std::println("{}", median(std::string{"Alamakota"}));
}
```

Uwagi do prowadzącego (W. Wysota):