Imię i nazwisko: ..... **S2** czas: 60 min

Przyjąć, że udostępniona jest przestrzeń nazw std, std::placeholders i boost

Zadanie 1 (7pkt)

Uruchomienie funkcji `zad1` prowadzi do wyścigu. Popraw kod klasy `Process`. Klasa ta symuluje wykonywanie zadań, wartość 0 składowej `token_` sygnalizuje brak zadania, `counter_` to licznik zadań. Dla dwóch wątków składowa `token_` jest około 2 razy większa, niż składowa `counter_`.

```
const int NUM = 10;
void zad1() {
   Process p1;
   Process p2(&p1, 1);
   p1.setIn(&p2);
   boost::thread thrd1( ref(p1) );
   boost::thread thrd2( ref(p2) );
   thrd1.join();
   thrd2.join();
}
```

Pytanie 1 (1pkt)

Zaproponuj zagadnienie, które Twoim zdaniem warto byłoby omówić na przedmiocie ZPR

```
class Process {
public:
   Process(Process* in = nullptr, int token = 0)
   : in_(in), token_(token), counter_(0) { }
   void setIn(Process* p) { in_ = p; }
   int get() {
      if(token_ > 0)
         ++counter_;
      int out = token_;
      token_ = 0;
      return out;
   }
   void operator()() {
      for(;;) {
         if(in_) {
            boost::mutex::scoped_lock lock1(inMutex_);
            boost::mutex::scoped_lock lock2(outMutex_);
            int x = in_->get();
            if( x > 0) {
               token_ = x + 1;
               if( counter_ > NUM )
                  break;
            }
         }
      }
   }
private:
   Process* in_;
   int token_;
   int counter_;
   boost::mutex inMutex_;
   boost::mutex outMutex_;
};
```

Zadanie 2 (5pkt)

Napisz szablon `replace_value`, który każdemu elementowi równemu `old` przypisze wartość `new_val`. Przykład użycia:

```
vector<int> v{1,2,3}; replace_value(v,1,-1); //teraz wektor ma elementy { -1, 2, 3 }
```

```
template<typename T>
void replace_value(T& c, const typename T::value_type& old, const typename T::value_type& new_val) {
```

Pytanie 2 (1pkt)

Ile godzin w semestrze poświęciłeś na przedmiot ZPR (wykład, projekt, kolokwia, nauka własna i inne)

Pytanie 3 (1pkt)

Podaj przykład danych, które wygodnie modelować grafem.

Zadanie 3 (3pkt)

Napis NAME zawiera Twoje nazwisko zapisane za pomocą wielkich liter ASCII (zamiast 'Ą' jest 'A'), np. WROZKA dla nazwiska Wróżka.

```
const string NAME = "␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣";
```

Podaj napis, który zostanie wydrukowany przez funkcję zad3

```
class Vis : public boost::static_visitor<void> {
public:
   Vis() {}
   void operator()(const int& i){ str_ += "i"; }
   void operator()(const std::string& s) { str_ += "s"; }
   string str_;
};
void zad3() {
   typedef boost::variant<int,string> V;
   std::vector<V> v;
   v.push_back(NAME); v.push_back(NAME.size()); v.push_back(v.front() ); v.push_back(v.back() );
   Vis vis;
   for_each(v.begin(), v.end(), [&](const V& v){apply_visitor(vis, v);});
   cout << vis.str_ << endl;
}
```

Zadanie 4 (4pkt)

Uzupełnij kod funkcji setParent, należy wołać metodę setParent dla każdego elementu wektora vec, przekazując adres tego wektora. Poniżej test.

```
typedef shared_ptr<Element> PElement;
typedef vector<PElement> VecElement;
class Element {
public:
   Element() : parent_(nullptr) {}
   const VecElement* getParent() const { return parent_; }
   void setParent(const VecElement* v) { parent_ = v; }
private:
   const VecElement* parent_;
};
void zad4() {
   vector<PElement> v;
   v.push_back( PElement(new Element()));
   setParent( v );
   assert( v[0]->getParent() == &v );
}
```

```
void setParent(VecElement& vec) {
```

Zadanie 5 (3pkt)

NAME zdefiniowano w zad. 3. Podaj napis drukowany przez funkcję zad5

```
int sum(int& s, int a)
{
  s += 2*a + 1;
  return s;
}
```

```
void zad5() {
   vector<int> v;
   v.push_back(1);
   v.push_back(NAME.size());
   int s = 0;
   for_each( v.begin(), v.end(), bind(sum, ref(s), bind(sum, ref(s), _1) ) );
   cout << s << endl;
}
```

Notatki / uwagi do prowadzącego