Imię i nazwisko: ..... **U2** czas: 60 min

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

Zadanie 1 (6pkt)

Poniżej przedstawiono klasę , która symuluje przetwarzanie zadań w różnych wątkach. Zadania są to obiekty typu `int` przechowywane w składowej `token`. Popraw kod, aby lepiej wykorzystywać wątki (popraw skalowalność rozwiązania).

```
void zad1() {
   Process p1(1);
   Process p2(2, &p1, 1);
   Process p3(3, &p2, 1);
   p1.setIn(&p3);
   boost::thread thrd1( ref(p1) );
   boost::thread thrd2( ref(p2) );
   boost::thread thrd3( ref(p3) );
   thrd1.join(); thrd2.join();
   thrd3.join();
}
```

Pytanie 1 (1pkt)

Zapisz wyrażenie regularne opisujące napisy, które reprezentują liczbę naturalną (bez zera).

```
boost::mutex global_mutex;
const int NUM = 10;
class Process {
 public:
 Process(int id, Process* in = nullptr, int token = 0)
 : id_(id), in_(in), token_(token), counter_(0) { }
 void setIn(Process* p) { in_ = p; }
 int get() {
  boost::mutex::scoped_lock lock(global_mutex);
  if(token_ > 0) {
   ++counter_;
  }
  int out = token_; token_ = 0;
  return out;
 }
 void operator()() {
  for(;;) {
   if(in_) {
    int x = in_->get();
    if( x > 0) {
     boost::mutex::scoped_lock lock(global_mutex);
     token_ = x + 1;
     if( counter_ > NUM ) {
      break;
     }
    }
 } } }
private:
 int id_;
 Process* in_;
 int token_; int counter_;
};
```

Zadanie 2 (6pkt)

Dostarczyć typ `my_string`, który jest napis nad alfabetem ASCII (obiekty typu `char`), podobnie jak `std::string`, ale operator porównania nie jest wrażliwy na wielkie i małe litery. Obok pokazano test. Funkcja standardowa `int toupper(int)` dostarcza argument przekształcony na wielką literę, funkcja standardowa `int tolower(int)` dostarcza argument przekształcony na małą literę, szablon `std::basic_string` ma parametry: typ znaku oraz klasę, wykorzystywaną m.in. do porównywania. Typowa deklaracja to: `typedef std::basic_string<char, std::char_traits<char> > string;`.

Drugi parametr szablonu ma metodę statyczną `static int compare( const char_type* s1, const char_type* s2, std::size_t count );` która zwraca 0 gdy napisy są równe, -1 gdy `s1` jest mniejszy niż `s2`, +1 gdy `s1` jest większy niż `s2`. Metoda ta jest wykorzystywana przez operator porównania dla napisu.

```
void zad2() {
   my_string s1 = "HELLO"; my_string s2 = "Hello"; my_string s3 = "hello";
   assert( s1 == s2 ); assert( s1 == s3 ); assert( s2 == s3 );
}
```

Zadanie 3 (3pkt)

NAME to Twoje nazwisko zapisane wielkimi literami ASCII (zamiast 'Ą' jest 'A'), np. `WROZKA` dla Wróżka.

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

Podaj napis, który zostanie wydrukowany przez funkcję `zad3`

```
typedef boost::variant<int,string> V;
int countNumLess(const std::vector<V>& v, int threshold) {
   class Vis : public boost::static_visitor<void> {
   public:
      Vis(int threshold) : threshold_(threshold), count_(0) {}
      void operator()(const int& i){ if(i < threshold_) ++count_; }
      void operator()(const std::string& s) { }
      int threshold_;
      int count_;
   } vis(threshold);
   for_each(v.begin(), v.end(), [&](const V& v){apply_visitor(vis, v);});
   return vis.count_;
}
void zad3() {
   std::vector<V> v;
   v.push_back("Abacki"); v.push_back(1); v.push_back(v.size());
   v.push_back(NAME.size()); v.push_back(NAME.size()/2); v.push_back(NAME.size()/4);
   cout << countNumLess(v, 3) << endl;
}
```

Zadanie 4 (4pkt)

Uzupełnij kod funkcji `getNamesEven`. Funkcja dostarcza nazwiska osób, które mają parzysty identyfikator.Dla pokazanego poniżej testu funkcja ta zwraca wektor zawierający napisy `"C", "D"`. Użyj algorytmu z biblioteki standardowej.

```
typedef std::pair<string, int> Person; //osoba: nazwisko i identyfikator
void zad4() {
   vector<Person> people = { Person("A", 1), Person("B", 3), Person("C", 2), Person("D", 4) };
   vector<string> names_even = getNamesEven(people);
   copy(names_even.begin(), names_even.end(), ostream_iterator<string>(cout,"␣") );
}
```

```
vector<std::string> getNamesEven(const vector<Person>& v) {
```

Zadanie 5 (3pkt)

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

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

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

Pytanie 2 (1pkt)

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

Pytanie 3 (1pkt)

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

Notatki / uwagi do prowadzącego