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

Przyjąć, że udostępniona jest przestrzeń nazw std

Zadanie 1 (5pkt)

Zmień implementację klas reprezentujących krawędzie i/lub wierzchołki, aby poprawnie przechowywać i zwalniać obiekty. Przykład użycia to funkcja `zad1`.

```
using PEdge = shared_ptr<Edge>;
using PWEdge = weak_ptr<Edge>;
using PVertex = shared_ptr<Vertex>;
using PWVertex = weak_ptr<Vertex>;

struct Edge {
  Edge(PVertex src, PVertex dest):src_(src),dest_(dest) {}
  ~Edge() {}
  PVertex src_;
  PVertex dest_;
};

struct Vertex {
  Vertex(int val) : val_(val) {}
  ~Vertex(){}
  int val_;
  vector<PWEdge> in_;
  vector<PWEdge> out_;
};
```

```
PVertex newVertex(int val) {
  return PVertex(new Vertex(val) );
}
PEdge newEdge(PVertex src, PVertex dest) {
  PEdge edge(new Edge(src, dest) );
  src->out_.push_back(edge);
  dest->in_.push_back(edge);
  return edge;
}

void zad1() {
  PVertex v1 = newVertex(1);
  PVertex v2 = newVertex(2);
  newEdge(v1, v2);
  cout << v1->out_[0]->src_->val_ << endl;
  cout << v1->out_[0]->dest_->val_ << endl;
}
```

Zadanie 2 (3pkt)

Podaj napis generowany przez `zad2` [ ], NAME to stała typu `std::string` zawierająca Twoje nazwisko zapisane wielkimi literami ASCII (zamiast 'Ą' jest 'A'), np. `WROZKA` dla Wróżka.

```
class E
  : public std::exception
{};
class B {
public:
  B(int i) : i_(i) {
    cout << i_;
  }
  virtual ~B() {
    cout << i_;
  }
  int get() const {
    return i_;
  }
private:
  int i_;
};
```

```
class D1 : public B {
public:
   D1(int i) : B(i+1) {}
};
class D2 : public B {
public:
   D2(int i) : B(i+2) {}
};
class M : public D1, public D2 {
public:
   M(int i) : D1(i), D2(i) {}
   int get() const {
     return D1::get() + D2::get();
   }
};
```

```
void f(int i, const M& m) {
  if(m.get() < 0) throw E();
  g(i+3);
}
void g(int i) {
  const int N = NAME.size() % 3;
  M m(N-i);
  f(i, m);
}
void zad2() {
  try {
    g(1);
  } catch(E&){
  }
  cout << endl;
}
```

Zadanie 3 (3pkt)

Napisz funkcję, która zwróci środkową wartość (medianę) dla 3 liczb.

```
int mediana(int a, int b, int c) {
```

Pytanie 1 (1pkt)

Dlaczego używa się różnych języków programowania do tworzenia tej samej aplikacji ?

Zadanie 4 (7pkt)

Analizowane obiekty są obrazami, obiektami typu `DescrPict`, albo opisami (tekstem), obiektami typu `DescrText`. Mamy do dyspozycji funkcje, które liczą podobieństwo:

```
double text2textSimilarity(const string& s1, const string& s2);
double text2pictSimilarity(const string& s, const Bitmap& b);
double pict2pictSimilarity(const Bitmap& b1, const Bitmap& b2);
```

Obiekty `Descr` dostarczają wzorca wizytatora i pozwalają pobrać obraz lub tekst.

```
class Visitor {
public:
   virtual void visit(const DescrPict& c) = 0;
   virtual void visit (const DescrText& r) = 0;
};
class Descr {
public:
   virtual void accept(Visitor& v) const = 0;
};
```

```
class DescrPict : public Descr {
public:
   virtual void accept(Visitor& v) const { v.visit(*this); }
   const Bitmap& getPicture() const;
};
class DescrText : public Descr {
public:
   virtual void accept(Visitor& v) const { v.visit(*this); }
   string getText() const;
};
```

Dostarcz implementację funkcji `similarity`.

```
double similarity(const Descr& d1, const Descr& d2 ) {
```

Zadanie 5 (6pkt)

Dostarczyć makro-komendę, klasę, która wykonuje przechowywane komendy w takiej kolejności, w jakiej je dodano. Interfejs komendy przedstawiono obok.

```
using PCmd = shared_ptr<Cmd>;
class Cmd {
public:
  virtual void operator()() = 0; //wykonuje komendę
  virtual void add(PCmd c) {} //dodaje komendę do kolekcji
  virtual ~Cmd() {}
};
```

```
class MacroCmd
```

Uwagi do prowadzącego: