Jedno z zadań na egzaminie będzie korzystało z rozwiązania poniższego zadania. Zachęcamy do jego samodzielnego rozwiązania. Wzorcowy kod znajduje się na kolejnej stronie. Powodzenia!

Zadanie przygotowawcze Napisz klasę Naukowiec zawierającą nazwisko, liczbę publikacji oraz cytowań naukowca. Klasa powinna zapewniać następujące operacje:

- konstruktor trójargumentowy inicjalizujący nazwisko, liczbę publikacji oraz cytowań,
- funkcję składową indeks zwracającą średnią liczbę cytowań na jedną publikację,
- funkcję składową publikuj zwiększającą liczbę publikacji o zadaną wartość,
- operator preinkrementacji ++ zwiększający o 1 liczbę cytowań naukowca,
- operator < porównujący naukowców. Lepszy jest naukowiec z większą liczbą cytowań, a jeżeli są takie same, to ten dla którego funkcja indeks zwraca większą wartość.
- operator « wypisywania danych naukowca do strumienia typu ostream oraz operator » wczytywania danych naukowca ze strumienia istream.

Przykład użycia klasy:

```
int main() {
   Naukowiec Bogdan("Bogdan",50,300), Czeslaw;
   cin >> Czeslaw;
   cout << Czeslaw.indeks();
   Bogdan.publikuj(3);
   if(Bogdan < ++Czeslaw)
      cout << Czeslaw << endl;
   else
      cout << Bogdan << endl;
}
```

Przykładowe rozwiązanie

```
class Naukowiec {
      string _name;
      int _publikacje, _cytowania;
   public:
      Naukowiec();
      Naukowiec (const string &name, int publikacje, int cytowania);
      double indeks();
      void publikuj (int n);
      Naukowiec &operator ++ ();
      friend bool operator < (Naukowiec &first, Naukowiec &second);
      friend ostream &operator << (ostream &stream, const Naukowiec &naukowiec);
      friend istream &operator >> (istream &stream, Naukowiec &naukowiec);
};

Naukowiec::Naukowiec() {}

Naukowiec::Naukowiec (const string &name, int publikacje, int cytowania):
   _name (name), _publikacje (publikacje), _cytowania (cytowania) {}

double Naukowiec::indeks() {
   return double(_cytowania)/_publikacje; }

void Naukowiec::publikuj (int n) {
   _publikacje += n; }

Naukowiec &Naukowiec::operator ++ () {
   _cytowania++;
   return *this; }

bool operator < (Naukowiec &first, Naukowiec &second) {
   return first._cytowania < second._cytowania ||
      (first._cytowania == second._cytowania && first.indeks() < second.indeks() ); }

ostream &operator << (ostream &stream, const Naukowiec &naukowiec) {
   return stream << naukowiec._name <<' '<< naukowiec._publikacje <<' '
      << naukowiec._cytowania; }

istream &operator >> (istream &stream, Naukowiec &naukowiec) {
   return stream >> naukowiec._name >> naukowiec._publikacje >> naukowiec._cytowania; }
```