

## Zadanie 9

---

Zdefiniuj klasę **Segment** reprezentującą odcinek  $[A, B]$  na osi liczbowej

```
class Segment {
    double A,B;
public:
    Segment(double A, double B) : A(A), B(B) { }
    // ...
};
```

Następnie zdefiniuj odpowiednie metody i funkcje, tak, aby dla odcinka `seg` i liczby `d` typu **double**

- wartością wyrażenia `d*seg` lub `seg*d` był odcinek powstały z `seg` przez przeskalowanie w stosunku `d` (tzn. współrzędne końca i początku tego odcinka mają być równe `d*A` i `d*B`, gdzie `A` i `B` to współrzędne początku i końca odcinka `seg`);
- wartością wyrażenia `seg/d` był odcinek powstały z `seg` przez przeskalowanie w stosunku  $\frac{1}{d}$  (odcinek `seg` „podzielony” przez `d`);
- wartością wyrażenia `seg+d` lub `d+seg` był odcinek `seg` przesunięty o `d` w prawo;
- wartością wyrażenia `seg-d` był odcinek `seg` przesunięty o `d` w lewo;
- wartością wyrażenia `seg1+seg2` był najmniejszy odcinek zawierający odcinki `seg1` i `seg2`;
- wartością wyrażenia `seg(d)` było **true** wtedy, gdy `d` należy do odcinka `seg` i **false** w przeciwnym przypadku.

Wszystkie te operacje nie powinny modyfikować swoich argumentów — powinny zwracać nowe obiekty.

Przeciąż też operator **operator<<** tak, aby następująca funkcja **main**

```
int main() {
    using std::cout; using std::endl;

    Segment seg{2,3}, s = 1 + 2*((seg-2)/2+seg)/3;

    cout << s << endl << std::boolalpha;
    for (double x = 0.5; x < 4; x += 1)
        cout << "x=" << x << ": " << s(x) << endl;
}
```

wydrukowała coś w rodzaju

```
[1,3]
x=0.5: false
x=1.5: true
x=2.5: true
x=3.5: false
```

---

*Termin: do 15 czerwca (włącznie)*

---

Rozwiązania, w postaci **jednego** pliku źródłowego zawierającego treść programu, proszę wrzucać w systemie EDU do katalogu „Foldery zadań / Zadanie\_XX”, gdzie 'XX' jest numerem zadania.

Nazwą pliku powinno być nazwisko z dużej litery (bez polskich znaków); rozszerzeniem musi być '.cpp', czyli np. `Malinowska.cpp`.