

Problema 1:

Scrieți codul necesar pentru ca următoarea funcție **main()**

```
int main() {
    //Rectangle r; // acest cod nu va COMPILA
    Rectangle r1 = { 10,10,20,20 };
    Rectangle r2 = { 15,50,25,15 };
    cout << "R1 = ";
    for (auto p : r1) cout << "Punct(" << p.x << "," << p.y << ") ";
    cout << endl;
    cout << "R2 = ";
    for (auto p : r2) cout << "Punct(" << p.x << "," << p.y << ") ";
    cout << endl;
    cout << "R1 e patrat: " << boolalpha << (bool)r1 << endl;
    cout << "R2 e patrat: " << boolalpha << (bool)r2 << endl;
    cout << "Suprafata lui r1 = " << (int)r1 << endl;
    cout << "Suprafata lui r2 = " << (int)r2 << endl;
    Rectangle r3 = r1 & r2;
    cout << "Dreptunghiul minim care include pe r1 si r2 este ";
    cout << " x=" << r3.GetTopLeftPoint().x;
    cout << " y=" << r3.GetTopLeftPoint().y;
    cout << " lungimea=" << r3.Width();
    cout << " latimea=" << r3.Height() << endl;
    return 0;
}
```

să afișeze pe ecran următoarele:

```
R1 = Punct(10,10) Punct(20,10) Punct(20,20) Punct(10,20)
R2 = Punct(15,15) Punct(25,15) Punct(25,50) Punct(15,50)
R1 e patrat: true
R2 e patrat: false
Suprafata lui r1 = 100
Suprafata lui r2 = 350
Dreptunghiul minim care include pe r1 si r2 este x=10 y=10 lungimea=15 latimea=40
```

Observatii:

- Parametri dați constructorului clasei Rectangle sunt doua colțuri opuse ale dreptunghiului (**NU** neapărat cel de stânga sus și cel de dreapta jos, și **NU** neapărat în aceasta ordine). Pentru exemplificare, R1 primește coordonatele stânga-sus și dreapta jos, iar R2 primește coordonatele stânga jos și dreapta sus.
- For each-ul iterează prin cele 4 colțuri ale dreptunghiului tot timpul în următoarea ordine: stânga sus, dreapta sus, dreapta jos, stânga jos și returnează prin iteratorul lui o structura Point cu doi membri x și y.
- Asigurați-vă ca specificațiile sunt respectate (ex: codul care nu ar trebui sa compileze, NU compilează).

Barem:

- **[5p]** Fiecare clasa este scrisa folosind un fișier heder și unul cpp. Codul din main este într-un fișier separat.
- **[3p]** operatorii de cast la bool si int din clasa Rectangle
- **[3p]** funcțiile care returnează lungime (Width) si latimea (Height)
- **[3p]** funcția GetTopLeftPoint() → care returnează punctul de stanga sus al dreptunghiului
- **[5p]** Constructorul de la clasa Rectangle
- **[5p]** operator & care calculează dreptunghiul minim care include doua alte dreptunghiuri
- **[4p]** funcțiile necesare din clasa Rectangle pentru funcționarea for-each-ului
- **[7p]** iteratorul pentru for-each