Задача 1: Банкова сметка

Условие

Да се създаде клас BankAccount със следните характеристики:

Полета/свойства:

- Owner (string) – собственик
- Balance (decimal) – баланс (не може да става отрицателен)

Конструктори:

- BankAccount(string owner) – създава сметка с баланс 0
- BankAccount(string owner, decimal initialBalance) – създава сметка с начален баланс (ако е отрицателен → приемете 0)

Методи:

- Deposit(decimal amount) – добавя сума (ако amount ≤ 0 → игнорира)
- Withdraw(decimal amount) – тегли сума; ако amount ≤ 0 или amount > Balance → не тегли и връща false, иначе тегли и връща true
- ToString() – връща текст: Owner: {Owner}, Balance: {Balance:F2}

Вход (от конзолата):

1. Име на собственик
2. Начален баланс
3. Цяло число n – брой операции
4. Следват n реда, всеки е:
    - deposit X или withdraw X

Изход:

- Накрая отпечатайте сметката чрез ToString().
- При неуспешно теглене отпечатайте: Insufficient funds

Примерен вход	**Примерен изход**
Ivan 100 5 deposit 50 withdraw 30 withdraw 500 deposit 20 withdraw 10	Insufficient funds Owner: Ivan, Balance: 130.00

Задача 2: Generic кутия (Box) с максимум

Условие

Да се създаде **шаблонен клас** Box<T> (generic), който пази **един елемент**.

Свойство:

- Value от тип T

Конструктор:

- Box(T value)

Метод:

- IsGreaterThan(T other) – връща true, ако Value е по-голямо от other.
    - За да работи сравнението, използвайте ограничение: where T : IComparable<T>

Задача:

- Прочетете от конзолата n (брой числа).
- За всяко число създайте Box<int> и го запазете.
- Прочетете още едно число x.
- Намерете колко от кутиите имат стойност **по-голяма от x** (чрез IsGreaterThan) и отпечатайте броя.

Примерен вход	**Примерен изход**
5 3 10 7 7 1 6	3 (по-големи от 6 са: 10, 7, 7)