In this exercise we study a simple abstract class.

Excercise 10 (Abstract class, 4p)

In this exercise you need to derive two classes from an abstract class and write a function that helps in testing the classes. Your starting point is an abstract class called Counter that defines an interface for incrementing and decrementing a counter and a conversion operator that can be used to read or print the current counter value.

Derive a class called LimitedCounter.

- The constructor takes two parameters: initial value and upper limit
- Counter can't be incremented over the upper limit. If inc() is called when counter has reached the upper limit nothing happens
- Counter can't be decremented below zero. If counter is zero then calling dec() will have no effect

Derive a class called OverflowCounter.

- The constructor takes two parameters: initial value and upper limit
- When counter has reached the upper limit incrementing the value will set the counter to zero.
- When counter is zero decrementing the counter sets counter to upper limit.

Implement function called UseCounter.

- `void UseCounter(Counter& ctr, int num);`
- Function takes two parameters: a reference to Counter and number that tells how many times the counter should be incremented or decremented. A negative value decrements counter and positive value increments counter.

Test your counters with different values and ways. Pay attention to the limits and make sure that they work properly.

Use the attached code as starting point.