Introductory Exercise warm-up 14: Stock Dynamics by hand

This little exercise will help you understand the principle of stock-driven models. It is **essential** that you understand this principle before you start programming this type of model in Python.

1. Inflow-driven model – 4 years sharp lifetime

In this example, we will assume a lifetime of exactly 4 years.
In an input-driven model, only the past inflows and the lifetime of each cohort are given.

Task: Calculate the missing data in the table below. Assume that the lifetime of the products in the stock is 4 years. While filling out the table, reflect on the following:

- What **operations** do you conduct to calculate the different dependent variables?
- In what **sequence** do you conduct the operations to fill out the entire table (e.g., column by column, row by row, or some kind of iteration…)?

	Annual stock (end of the year)	**Stock change** (during the year)	**Inflow** (during the year)	**Outflow** (during the year)
Year t	**S_t [tons]**	**dS_t/dt [tons/yr]**	**I_t [tons/yr]**	**O_t [tons/yr]**
t_0=2000	0	0	0	0
2001			100	
2002			200	
2003			400	
2004			800	
2005			1600	
2006			3000	
2007			4000	
2008			5000	
2009			5500	
2010			6000	
2011			6000	
2012			6000	

Stock-driven models

A stock-driven model is a model in which the stock (measured at the end of the accounting year) and the lifetime of each cohort are given; both as independent variables. Subsequently, the stock change, the inflow, and the outflow are calculated (dependent variables).

Task: Calculate the missing data in the table below. Assume that the lifetime of the products in the stock is 4 years. While filling out the table, reflect on the following:

- What **operations** do you conduct to calculate the different dependent variables?
- In what **sequence** do you conduct the operations to fill out the entire table (e.g., column by column, row by row, or some kind of iteration…)?

	Annual stock (end of the year)	**Stock change** (during the year)	**Inflow** (during the year)	**Outflow** (during the year)
Year t	**S_t [tons]**	**dS_t/dt [tons/yr]**	**I_t [tons/yr]**	**O_t [tons/yr]**
t_0=2000	0	0	0	0
2001	100	100	100	0
2002	200			
2003	400			
2004	800			
2005	1600			
2006	3000			
2007	4000			
2008	5000			
2009	5500			
2010	6000			
2011	6000			
2012	6000			

Next step: how to write these algorithms in Python, using a for loop?