1. Create a class called Tank that can be filled and emptied (let’s say it has acts like a stack), and has a termination condition that it must be empty when the object is cleaned up. Implement IDisposable interface. Dispose() method should check this termination condition. In main(), test the possible scenarios that can occur when your Tank is used. **[2 points]**
2. Write in a file 4 integers and then reread the third and fourth ones using random access to the file. **[3 points]**
3. Read an arbitrary file, compress it to another file and display the compression ratio. **[3 points]**
4. Create an application that reads the information of Customers from a text file. The data that is available for each person is:

* Name, Username, password
* Address
* Hotel reservation
  + City
  + Hotel Name
  + Check-in date
  + Check-out date

Following the Repository design pattern (resources available in the reading list file):

* Read the Customers information from the file
* Create the necessary classes
* Implement the following methods:
  + Customer[] FindByName(string customerName)
  + Reservations[] GetCustomerRezervations(string customerId)

**[7 points]**

Bonus:

Create the EncriptPassword and DecryptPassword methods for protecting user security

**[2 points]**