**Reflection, Interfaces and Basic Recursion**

[***Recursion***](https://en.wikipedia.org/wiki/Recursion) occurs when something is defined in terms of itself.

For example the famous [Fibonacci](https://en.wikipedia.org/wiki/Fibonacci_number) sequence is defined recursively like this:

F(n) = F(n-1) + F(n-2)

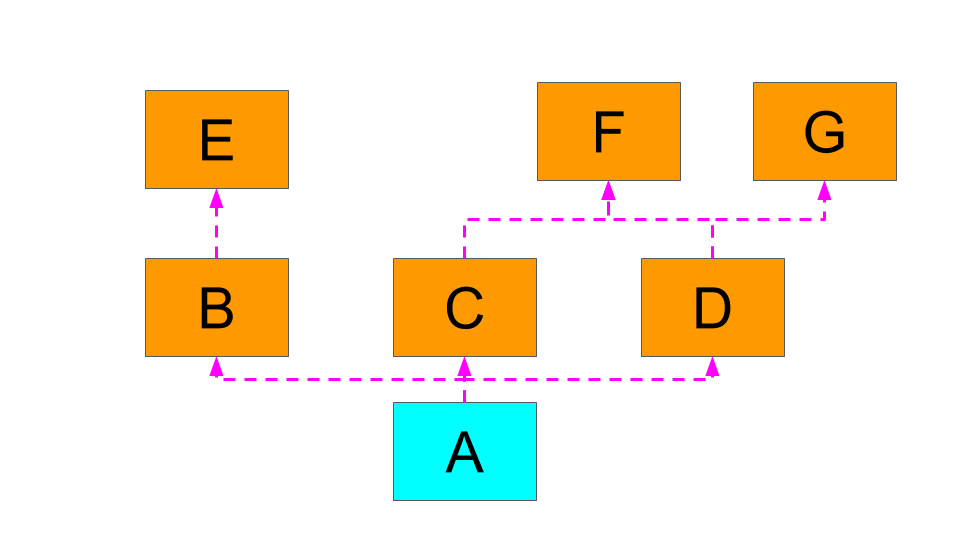
where:

F(0) = 0, F(1) = 1.

Recursion is a very useful way to solve problems in computer science and write algorithms in a compact and elegant way.

In this exercise we will use [***Recursion***](https://en.wikipedia.org/wiki/Recursion)and **Java Reflection** to solve the problem of **"Finding all Implemented Interfaces of a class".**

Example:



(***B, C, D, E, F, G*** are interfaces)

The class **A**implements the interfaces **B**, **C** and **D**.

However because:

* Interface **B** extends**E**
* Interface **C** extends**F**
* Interface **D** extends **F** and**G**

***"All the implemented interfaces of the class A" are:***

***{B, C, D, E, F, G}***

Using Java Reflection and Recursion, implement the following method to return all the implemented interfaces of a given class: