Contracts (Ada 2012)

The purpose of this exercise is to write contracts for a Stack package.

Question 1 (See sources in ex1 directory)

Build and run the application.
What is wrong here?
What kind of contracts do we already have?

Question 2 (Copy ex1 to ex2 directory)

Incrementing or decrementing Length can cause out of range exceptions.

Declare two functions Is_Full and Is_Empty and use preconditions to guard against that.

Check that you get the expected exception when a precondition is false [hint : activate the assertion using -gnata switch]

Question 3 (Copy ex2 to ex3 directory)

Now we are protected against errors in the library.

What about client code ? Can we call Pop after Push after Reset ?

Use Is_Full and Is_Empty to write postconditions.

Question 4 (Copy ex3 to ex 4 directory)

Now client code can be sure Pop can be called after Push but not after Reset. Still, this is not enough, postconditions are too weak to ensure absence of error in client code in general.

Introduce a new function Size and use it to specify Pop and Push using 'Old attribute.

[Hint: Size returns the number of elements in the stack. The Post condition of Pop will describe how it affects the size. Likewise for Push].

Question 5 (Copy ex4 to ex5 directory)

Now we have properly specified when each of our procedure can be used. But we still haven't got full functional correctness. Indeed, we do not know what is the value returned by Pop. To encode that, add a Get_Model function that returns an array representing the internal value of the stack.

[Hint: Straitforward. Get_Model will describe a simple view of the stack]