

Contracts (Ada 2012)

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

Note : Follow the instructions, keep the structure of the folders, do not rename files, use already provided functions and macros, do not rename provided functions and macros

Question 1 (Go to ex1 folder)

Build and run the application using Alire.

```
alr build
```

```
alr run
```

What is wrong here? What kind of contracts do we already have?

Provide your answer in the file answers.txt in the ex1 folder

Question 2 (Go to ex2 folder)

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. Is_Full and Is_Empty should be declared in stack.ads using expression functions.

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

Note : Do not edit stack.adb

Question 3 (copy ex2 folder to ex3 folder)

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.

Note : Do not edit stack.adb

Question 4 (copy ex3 folder to ex4 folder)

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. Size should be declared in stack.ads using an expression function.

[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].

Note : Do not edit stack.adb

Question 5 (copy ex4 folder to ex5 folder)

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.

Get_Model should be declared in stack.ads using an expression function.

[Hint: Straitforward. Get_Model will describe a simple view of the stack as an array containing the element of the stack]

Note : Do not edit stack.adb