

The University of New South Wales
SENG2011: Workshop on Reasoning about Programs
Sample Mini-Examination
(First exercise from 2021 Supplementary Examination)

- Two questions: total time allowed: 60 minutes

ex1.dfy

0 marks

It is a property of natural numbers that the product of two consecutive natural numbers (CNNs) is even. In other words, for all $n \geq 0$, $n * (n + 1)$ is even.

Write a lemma with signature:

```
lemma LemCNN(n: nat)
```

that proves this property. You may use any proof technique you like, but you must justify every step in the proof (hence Level 3). This lemma should allow Dafny to assert this property for the general case of all natural numbers.

Limit the time you spend on each exercise.

Submission: **give se2011 sample4 ex1.dfy** (command will not work before the exam)

ex2.dfy

0 marks

Write a method that returns with the **index** of the first triple of consecutive, identical elements in an array of integers. This method should have signature:

```
method GetTriple(a: array<int>) returns (index: int)
```

If there is no triple in the array then the method returns the length of the array. The array may be any length. You may assume that there is *at most one instance* of a triple in the array.

The index of a triple is simply the index of the first of the three elements. For example, if the array is `[0,42,42,42]` then the returned index is 1 as the triple 42 starts at index 1. If the array is `[1,1,2]` then the returned value is 3 because there is no triple and the length of the array is 3. A sample testcase is:

```
a := new int[4] [0,42,42,42];  
assert a[0] == 0 && a[1] == a[2] == a[3] == 42;  
b := GetTriple(a);  
assert b==1;      // index of the first element in the triple
```

Limit the time you spend on each exercise.

Submission: **give se2011 sample4 ex2.dfy** (command will not work before the exam)

End of Short Sample Examination