

Question 5

Part A

CODE:

```
1 // http://rise4fun.com/Dafny/Is4L
2 // return the index of an integer -- key in an integer array -- a
3 // if the key is not found then return -1
4
5 method FindWithLoops(a: array<int>, key: int) returns (i: int)
6 requires a != null;
7 ensures i >= 0 <==> 0 <= i <= a.Length - 1 && a[i] == key;
8 {
9     i := a.Length - 1;
10    while i >= 0
11        invariant forall j :: a.Length - 1 > j > i >= 0 ==> a[j] != key;
12    {
13        if a[i] == key {
14            return;
15        }
16        i := i - 1;
17    }
18 }
```

PROOF:

$$I = \forall j, a.Length - 1 > j > i \geq 0 \implies a[j] \neq \text{key} \quad (1)$$

At the beginning of the loop there is no possible j so the invariant holds.

During the loop if there exists $j > i$ such that $a[j] = \text{key}$ then there must have been an i^{th} iteration of the loop where $a[i] = \text{key}$ and therefore the function would have returned at line 18 which is a contradiction.

At the termination of the loop there are two cases

$$\exists i, a.Length - 1 > i \geq 0 \wedge a[i] = \text{key}$$

where the key was found or,

$$i = -1 \implies \nexists i, a.Length - 1 > i \geq 0 \wedge a[i] = \text{key}$$

in which cases the invariant also holds since

$$\forall j, a.Length - 1 > j > i = -1, \nexists a[j] = \text{key}$$

Therefore the invariant holds before during and after the loop.