Assignment 1: Proving Correctness

Below are a number of simple programs. Type annotations are removed to make the code less verbose. All variables are of type `int` or `int array`. We want to verify that these programs are 'correct'. The informal intent of the programs are stated below. Give first formal specifications for them. If you think the informal specifications are not precise enough (and probably they are not), decide for yourself what you think they should mean.

For at least the first three programs, give the steps to proceed in order to verify their correctness with respect to your formal specifications. Try to identify which rules or transformations are to be used to formally handle each of your steps. Take closer a look at these rules or transformations, and check if you understand the idea behind them and their mechanics.

1. A GCL program to swap two elements of an array:

```
$swap(a, i, j \mid a')\{$
  var $tmp$ in
  $tmp := a[i]$ ; $a[i] := a[j]$ ; $a[j] := tmp$ ; $a' := a$
  end }
```

2. A GCL program to return an index pointing to a minimium element of an array segment $a[i..N)$. The segment is assumed to be non-empty ($N > i$).

```
$minind(a, i, N \mid r)\{$
  var $min$ in
  $min,\ r\ :=\ a[i],\ i$ ;
  while $i<N$ do {
    if $a[i] < min$ then $min,\ r\ :=\ a[i],\ i$ else skip
    $i$++
    }
  end }
```

3. A GCL program to ascendingly sort an array segment $a[0..N)$.

```
$sort(a, N \mid a')\{$
  var $i$ in
  $i := 0$ ;
  while $i < N-1$ do {
    $m := minind(a, i+1, N)$ ;
    if $a[m]<a[i]$ then $a := swap(a, i, m)$ else skip ;
    $i$++
    }
  $a' := a$
  end }
```

4. Another GCL program to ascendingly sort an array segment $a[0..N)$. This program implements a very simple algorithm. It non-deterministically picks two consecutive elements in the segment, that is still 'out of order', and swaps them. This is repeated until there are no more such elements. The invariant of this program is perhaps rather trivial, but proving its termination can be more challenging.

```
$ndsort(a, N \mid a')\{$
  while $(\exists i : 0 \leq i < N-1 : a[i] > a[i+1])$ do $\{$
    var $i$ in
    assume $0 \leq i < N-1 \ \wedge \ a[i] > a[i+1]$
    $a := swap(a, i, i+1)$
    end
    $\}$
  $a' := a$
  $\}$
```