* Proof of correctness for Selection Sort

```
function SelectionSort(array):
        for i from 0 to length(array) - 1 do:
            minimum_index = i
        for j from i+1 to length(array) do:
            if array[j] < array[minimum_index] then:
        minimum_index = j
      swap array[i] with array[minimum_index]
end for
end function
```

→ loop Invariant: At the begining of each iteration the sub array array[0 : i] contains smallest i element from entire array

Base case: 1st iteration, the subarray array[0:0] is empty, it is sorted, the loop invariant is valid at this stage.

Inductive step: for first i iteration. This implies that the subarray array[0:i] consist of smallest element. So loop invariant holds true.

iteration i+1 :- the algorithm scans the unsorted postion of array [i:n] to locate smallest element. After swapping, the element array [i] is the smallest element

∴ So array [0:i+1] contains smallest i+1 element, So loop invariant continues to hold true after i+1th iteration

Termination :- when i=n-1 loop completetes execution the array is sorted

Conclusion :- Since the loop invariant hold true throughout the algorithm execution. The algorithm gurantee a sorted output for any valid input array.