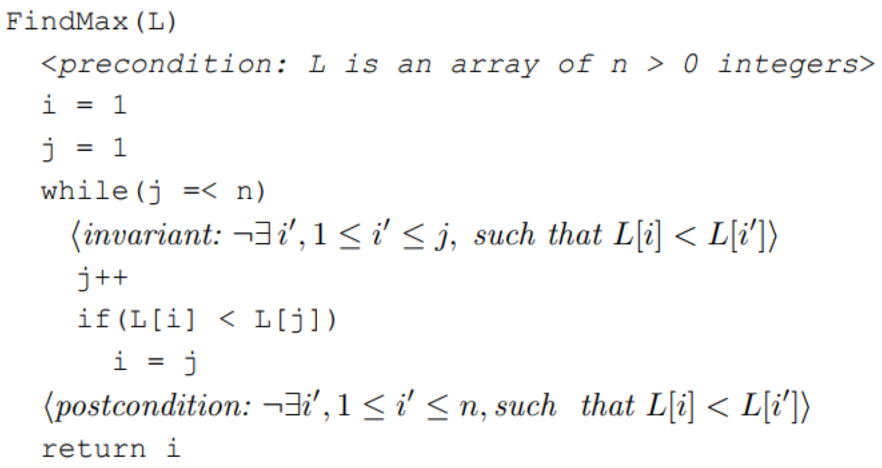
# 1 Loop invariant



```
FindMax(L)
  <precondition: L is an array of n > 0 integers>
  i = 1
  j = 1
  while(j =< n)
    ⟨invariant: $\neg\exists\, i', 1 \leq i' \leq j,$ such that $L[i] < L[i']$⟩
    j++
    if(L[i] < L[j])
      i = j
  ⟨postcondition: $\neg\exists i', 1 \leq i' \leq n,$ such that $L[i] < L[i']$⟩
  return i
```

To establish a loop invariant we need to show two things:

- **initialisation**: The invariant is true prior to the first iteration of the loop
- **maintenance**: If the invariant is true before an iteration of the loop, it remains true in the next iteration

If those two properties hold, the invariant is true prior to every iteration of the loop and when the loop terminates. Typically we'd use code in the loop body to prove that the invariant remains true before each iteration.

## 1.1 Proving partial correctness using while

To use the while rule ina proof of correctness of *FindMax*

$$\psi \text{ is } \neg\exists \text{i'},1 \leqslant \text{i'} \leqslant j \text{ such that } L[i] < L[\text{i'}]$$

$$B \text{ is } j \leqslant n \text{ and } \neg B \text{ is } j \nleqslant n$$

B holds because of precondition $n > 0$ and $j$ is initialised to 1. So provided the loop therminates $\psi$ and $\neg B$ together imply that

$$\neg\exists \text{i'},1 \leqslant \text{i'} \leqslant n \text{ such that } L[i] < L[\text{i'}]$$