Pre- and Post-Conditions

PRE: what needs to be true for code to do its job
POST: what the code does

sum(A) PRE: A is an array of summable items
POST: returns $\sum_{i=0}^{len(A)-1} A[i]$

sort(A) PRE: A is an array of of comparable items
POST: $A[0] \le A[1] \le \dots \le A[len(A)-1]$
and A is permutation A_{in}

sort(A)
for i=0 to len(A)-1
  A[i] = i

this code satisfies, but doesn't sort, need 2nd part of POST

$sq \leftarrow a * a$ PRE: a is multipliable
POST: $sq = a * a$

$min \leftarrow -b / (2 * a)$ PRE: arithmetic is possible and $a \neq 0$
POST: min = result of arithmetic

$x \leftarrow A[i] + A[j]$ PRE: $0 \le i, j < len(A)$ and elts are summable
POST: x = result of arithmetic

Invariants

Loop Invariant : something true at start of every iteration (right before condition test)

Loop Invariant Thm:

the loop invariant

For predicate P, if
a) P is true when loop 1st starts (after 0 iterations)
b) whenever P is true before an iteration and condition (guard) is true P is true after next iteration

then P is a loop invariant (T before testing condition after n iterations for all n s.t. there are n iterations)

iteration

Also want loop terminates
P true at termination and condition is false → postconditions are met

```
sum(A)
  total ← 0
  i ← 0
  while i < len(A)
    total = total + A[i]
    i ← i + 1
  return total
```

INU : $i = n$

$total = \sum_{k=0}^{i-1} A[k]$

Base case ($n = 0$) : $i = 0$ by initialization so $i = n$

$total = 0$ by initialization

$\sum_{k=0}^{i-1} A[k] = \sum_{k=0}^{-1} A[k] = 0 = total$

Term : if $n = len(A)$ then
$i = len(A)$ and loop terminates
if $n < len(A)$ then
$i < len(A)$ and loop continues

∴ loop terminates w/ $n = len(A)$

Ind : Suppose INV true after n iterations and $i < len(A)$
[want : INV true after n+1 iters]

[want $i_{new} = n+1$]

$i_{old} = n$

$i_{new} = i_{old} + 1 = n+1$

$total_{old} = \sum_{k=0}^{i_{old}-1} A[k]$

$total_{new} = total_{old} + A[i_{old}]$

$= \sum_{k=0}^{i_{old}-1} A[k] + A[i_{old}]$

$= \sum_{k=0}^{i_{old}} A[k]$

$\sum^{i_{old}}$ ...

Post: Suppose loop terminates.
Then $n = len(A)$
and $i = len(A)$

and $total = \sum_{k=0}^{i-1} A[k]$

$= \sum_{k=0}^{len(A)-1} A[k]$ POST

which is returned

$$= \sum_{k=0} A[k]$$

$$= \sum_{k=0}^{i_{new}-1} A[k]$$