EXAMPLE OF PROVING LOOP CORRECTNESS

```
for (k=0; k<10; ++k)
{
    sum += x;
}
```

You will be given the loop.

Precond: $k=0$, $sum=0$

☆ Inv(k): sum equals $x*k$ after the k^{th} iteration

Postcond: $sum = 10*x$ when done

You will be given info about the loop.

Verify 4 things

① Basis property – does the Precond imply Inv(0) (loop invar initially true)

Yes. Sum is initially 0.

② Inductive. Does $Inv(k) \rightarrow Inv(k+1)$? (stays true)

$sum_k = kx$

$sum_{k+1} = sum_k + x$

$= kx + x$

$= (k+1)x$ ✓

③ Finite? Yes. k eventually 11.

④ After 10 iter plug into loop Invar to see if post cond satisfied.

$Inv(10) =$ sum equals $10x$

which is Post cond.

```
sum = 0;
for (i = 0; i < 100; ++i)
    sum += a[i];
```

Precond : sum is 0

Postcond : sum is sum all 100 ele

Inv (k) : after k iter,
"sum" equals sum of 1st k elements of array

---

(1) Basis property

does the precond ⟶ Inv(0)

(precond: sum = 0; Inv(0): we haven't summed anything.)

(2) Inductive step

After k iter $\quad sum_k = \sum_{i=0}^{k-1} a[i]$

what about sum_{k+1} ?

$$= sum_k + a[k]$$

$$= \sum_{i=0}^{k} a[i]$$

(Inv (k+1).)

(3) Finite? i does reach 100

(4) When done, k = 100 iter

Inv(100) $\quad sum = \sum_{i=0}^{99} a[i]$:)

ex. 4.5.6

```
exp = 1;
for (i = 0; i < m; ++i)
    exp *= x;
```

Precond: exp is 1.
Postcond: exp is x^m.
Invariant: After the k^{th} iteration, $exp = x^k$.

---

(1) Basis. Does the precondition imply Inv(0)

$exp = 1 \longrightarrow exp = x^0$

yes, they are the same

(2) Inductive Does $Inv(k) \rightarrow Inv(k+1)$

After k iterations $exp_k = x^k$

on k+1 iter, we multiply by x.

$$exp_{k+1} = exp_k * x$$
$$= (x^k) * x$$
$$= x^{k+1} = Inv(k+1)$$

(3) Loop terminates. ✓

(4) Does Inv(m) imply postcond?

$exp = x^m$ is post!

4.5.9

```
a = initial;
while (a > 0)
    a = a - 2;
```

Precond: a equals initial, and initial ≥ 1.

Postcond: a equals 0 if initial is even
−1 if initial is odd

Inv (k): $a \geq -1$ and either
both a and initial are even, or
both a and initial are odd

---

(1) Basis. Does precond imply Inv (0) ?

$a = initial$,
$initial \geq 1$
$a \geq 1$

Since $a \geq 1$, $a \geq -1$.
Since $a = initial$,
both same parity.

(2) Inductive. Does Inv (k) → Inv (k+1)?

If a even, $a-2$ still even
If a odd, $a-2$ still odd

The only way we actually perform a $(k+1)^{st}$ iteration is if $a \geq 1$ after k iterations.

⇒ After $(k+1)^{st}$ iteration, $a \geq -1$.
(because $a_{k+1} = a_k - 2$)

(3) Loop terminates since a eventually becomes no longer > 0 since we keep subtracting 2 from it.

(4) Does Inv (last iteration) imply Postcond?

After last iteration, $a > 0$ is false and $a \geq -1$ is true } $a = 0$ or -1

a and initial always had same parity
even : $a = 0$
odd : $a = -1$