Some intuition for loop invariants. When we are at the $j$th iteration of a loop, the inductive hypothesis states what the value of some variable is up to that point. We always assume what the results are just before we begin to compute the procedure at $j$. The inductive step is to prove what the results are just before the $j + 1$st iteration, which means we must find the results after computing the procedure at $j$.

**1.** Let $E$ be the expression $n - i$. With each iteration of the loop `i` increases by 1. Therefore $n - i$ decreases by 1 with each pass. Eventually $E$ will be negative, and the loop will terminate. In particular, when $n - i \leq -1$.

We prove the following statement by induction on the value of the variable `i`.

STATEMENT $S(m)$: If we reach the loop test $i \leq n$ with the variable `i` having the value $m$, then the value of the variable `sum` is $m(m-1)/2$.

BASIS. The basis is $m = 1$. When we first enter the loop we reach the test with `i` having value 1 and `sum` having value 0. We see that $1(1-1)/2 = 0$. So the basis is proven.

INDUCTION. Assume $S(m)$. We shall prove $S(m+1)$.

We assume here that we are not entering the loop for the first time. If $m > n$, then when `i` has the value $m$ we do not reach the loop test. Thus with `i` having the value $m+1$, we do not reach the loop test. In that case $S(m+1)$ is trivially true.

If $m \leq n$, then we consider what happens when we execute the body of the loop with `i` having the value $m$. By the inductive hypothesis, `sum` has value $m(m-1)/2$ and `i` has value $m$ (yes we repeat that `i` has value $m$ again). After the body of the loop is executed, and when we reach the loop test, `sum` has the value $m(m-1)/2 + m = m(m+1)/2$ and `i` has the value $m+1$. We have proven $S(m+1)$, therefore $S(m)$ holds for $m \geq 1$.

We expressed earlier that the loop will terminate when $n - i \leq -1$. That is, when `i` has the value $n+1$. Thus after the body terminates $S(n+1)$ must hold, because we reach the test loop when `i` has at most the value $n+1$. This statement says that `sum` has the value $m(m+1)/2$, which is the desired result of the program. ✦

The basis gives us the result upon entering the loop, where `i` has value 1 and `sum` has value 0. Afterward, we know that when `i` takes on $n+1$ we reach the loop test and fail. Hence when `i` has value $m$ we know that we reach the loop test once when $m > n$ and fail. There is no point in this process where $m > n$ and we do not reach the loop test, so this can be omitted.

**2.** We prove the following statement by induction on the value of the variable `i`.

STATEMENT $S(m)$: If we reach the loop test $i < n$ with the value of variable `i` being $m$, then the variable `sum` has the value $\sum_{i=0}^{m-1} A[i]$ where $A$ is an array of integers.

BASIS. The basis is $S(0)$. When we enter the loop `i` has the value 0 and `sum` is $\sum_{i=0}^{-1} A[i] = 0$. Thus the basis is true.

INDUCTION. Assume $m \geq 0$ and $S(j)$ is true for $0 \leq j \leq m$. We shall prove $S(m+1)$.

Assume $m < n$. The inductive hypothesis states that `sum` has the value $\sum_{i=0}^{m-1} A[i]$. After executing the body of the loop, `sum` has value

$$\sum_{i=0}^{m-1} A[i] + A[m] = \sum_{i=0}^{m} A[i]$$

and `i` has value $m+1$. This proves the inductive step.

The loop terminates when $m = n$. But we reach the loop test at this value. Hence $S(n)$ holds, and this statement claims that the value of `sum` is $\sum_{i=0}^{n-1} A[i]$ after executing the program. That is, `sum` has the value of the integers `A[0..n-1]`, which is the desired result. ✦

I realize long after that the exercise did not require a proof by induction, just the loop invariant.

**3.** We prove the following statement by induction on the value of the variable `i`.

STATEMENT $S(k)$: If we reach the loop test $i \leq n$ with `i` having the value $k$, then `x` has the value $2^{2^{k-1}}$.

BASIS. The basis is $k = 1$. Then we reach the loop test with `i` being 1 and `x` being $2^{2^0} = 2^1 = 2$, which is what the procedure assigned to `x` intially. Thus the basis is proven.

INDUCTION. Assume $k \geq 1$ and $S(j)$ is true for $1 \leq j \leq k$. We shall prove $S(k+1)$.

The inductive hypothesis states that `i` is $k$ and `x` is $2^{2^{k-1}}$. We compute the instruction in the loop. This assigns to `x` the value

$$2^{2^{k-1}} \times 2^{2^{k-1}} = 2^{2^{k-1}+2^{k-1}} = 2^{2(2^{k-1})} = 2^{2^k}$$

and `i` the value $k+1$. This proves the inductive step.

The loop terminates when $k > n$, and this is exactly when $k = n+1$. Hence $S(n+1)$ holds. That is, when the loop terminates, `x` has the value $2^{2^n}$. ✦

**4.** We state an appropriate loop invariant with argument $n$, the number of times the body of the loop has been executed.

STATEMENT $S(n)$: If we reach the loop test $x \geq 0$ then `sum` has the value of the sum of the sequence $x_1, x_2, \ldots, x_{n-1}$.

The loop test fails when $x < 0$, and thus is after executing the loop $n$ times. Hence we reach the loop test at most at the $n+1$st iteration. So $S(n+1)$ holds. This states that `sum` has the value of the sum of the sequence $x_1, x_2, \ldots, x_n$.

**5.** Let `f` be the procedure that computes the program fragment. We find that `f(13)` $>$ `f(14)`, so $n = 13$ is the largest value the procedure gives a correct output for. An implication is that induction has a limit when we continually increment numbers. Another is that mathematically sound results may not always be applicable to computers.

**6.**

STATEMENT $S(n)$: If we have gone around the loop $n \geq 1$ times, then $j > i + 1$.

BASIS. The basis is $n = 1$. At the start, $j = i + 1$. After the body of the loop is executed then $j$ is incremented. Thus $j > i + 1$, hence the basis is true.

INDUCTION. Assume $n \geq 1$ and $S(n)$ is true. We shall prove $S(n + 1)$. By the inductive hypothesis we have gone around the loop $n$ times, and we know that $j$ is incremented after each time. But we have gone around the loop $n + 1$ times, so $j = n + 1 + i + 1$. Clearly $j > i + 1$, because $n \geq 1$ (and $n + 1 \geq 1$). The inductive step is proven.

I suppose we do not need to do complete induction on them all.