Principle of Mathematical Induction
Suppose that $S(n)$ is some statement about the integer $n$ and we wish to prove that $S(n)$ is true for all positive integers $n$. Then,

1. Prove that $S(1)$ is true;
2. Prove that if $S(n)$ is true (the Inductive Hypothesis—IH), then $S(n+1)$ is true.
Example: Prove $1 + 2 + \ldots + n = \frac{(n)(n+1)}{2}$

Proof:

1. Show true for $n = 1$.
   Let $n = 1$, then $(1)(1+1) / 2 = 2/2 = 1$

2. Given: $1 + 2 + \ldots + n = \frac{(n)(n+1)}{2}$, show
   $1 + 2 + \ldots + n + (n+1) = \frac{(n+1)((n + 1) + 1)}{2}$

   $(1 + 2 + \ldots + n) + (n + 1)$
   $= \left[\frac{n(n+1)}{2}\right] + (n + 1)$, by the IH
   $=\left[\frac{n^2 + n}{2}\right] + (n + 1)$
   $=1/2[n^2 + 3n +2]$
   $= \frac{1}{2}[(n + 1)(n + 2)]$
   $= \frac{(n + 1)[(n + 1) + 1]}{2}$

   ---------------------------------
Homework 1:
Is $2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$ (start with $n = 0$)?

Homework 2:
Is $\frac{1}{1 \times 2} + \frac{1}{2 \times 3} + \ldots + \frac{1}{(n-1) \times n} = (3n - 2)/(2n)$ (start with $n = 1$)?
Some Program Correctness Definitions

LOOP INVARIANTS

A loop invariant must satisfy the following conditions:
- It must be true on loop entry.
- It must be independent of the number of loop traversals.
- It must imply the desired condition on exit.
CORRECTNESS DEFINITIONS
Let A be an assertion describing any assumptions about the data for a (flowchart) program and let C be an assertion describing what the program is supposed to accomplish (i.e., the correctness assertion). Then the program is said to be partially correct with respect to A and C provided that whenever the program is executed with data satisfying assumption A, if the program terminates, C is true.
A (flowchart) program is totally correct (with respect to A and C) provided it is partially correct (with respect to A and C) and it terminates for all values that satisfy assumption A.
METHOD OF INDUCTIVE ASSERTIONS

1. Attach assumption A to the beginning of the program and attach the assertion C to the terminal point of the program.

2. In addition, discover and attach assertions (which describe relationships concerning the values of the variables) to some other points in the program. In particular, attach an assertion to at least one point in every closed path (loop) in the program.

3. Then prove that for every path in the program leading from point i, which has assumption $A_i$ attached to it, to a point j, which has assumption $A_j$ attached to it (with no intervening assumptions attached to points from i to j), that if execution is at point i and assertion $A_i$ is true, and execution proceeds from i to j, then when execution reaches point j, the assumption $A_j$ will be true.
Example: Prove the following flow chart calculates $M!$ (i.e., $1 \times 2 \times \ldots \times M$).
Proof:
(i): First time execution reaches point 1, \( I = 2 \) and \( J = 1 \),
\( J = (2 - 1)! = 1! = 1 \), so \( 1 = 1 \).

(ii): Assume \( J_n = (I_{n-1})! \) is true the \( n^{\text{th}} \) time execution
reached point 1, and show
\[ J_{n+1} = (I_{n+1} - 1) \]
is true the \( (n + 1)^{st} \) time execution
reaches point 1.

Then, \( J_n = (I_n - 1)! \) by the Inductive Hypothesis,
and \( I_{n+1} = I_n + 1 \), or \( I_n = I_{n+1} - 1 \).
Then \( n + 1^{st} \) time execution reaches point 1,
\[ J_{n+1} = J_n I_n = (I_{n-1})! \] \( I_n \) (by the Inductive Hypothesis),
so \( J_{n+1} = I_n! = (I_{n+1} -1)! \) because \( I_n = I_{n+1}-1 \).
Must also show $J = M!$ at termination (i.e., eventually execution reaches point 2 with $I > M$ and $J = M!$). Because $M \geq 1$ and $I \geq 2$ and $M$ does not change, $I$ will eventually exceed $M$. If $M = 1$, $I > M$ the first time and $J = 1$, which is $1!$. Otherwise, execution proceeds until $I - 1 = M$, or $I$ exceeds $M$ by 1. Because $J = (I - 1)!$, $J = M!$

Therefore, QED quod erat demonstrandum (KWAWD eh-RAHT dem-on-STRAHND-um) “That which was to have been proved.” Traditionally placed at the end of proofs.
**Homework:** Prove the following flowchart computes $M \times N$ (The product of $M$ and $N$), provided $M$ and $N$ are integers and $M \geq 0$. You must discover a loop invariant and a correctness assertion first. What happens if $M \leq 0$?
Start

Get M, N

I := M
J := 0

I = 0

T

Stop

J := J + N
I := I - 1