1. Create a full proof outline of the following formal proof.

$1.\{T\}\ k \coloneqq 0\ \{k = 0\}$	forward assignment
$2.\{k = 0\}\ x \coloneqq 1\ \{x = 1 \wedge k = 0\}$	forward assignment
$3.\{T\}\ k \coloneqq 0; x \coloneqq 1\ \{x = 1 \wedge k = 0\}$	sequence 1,2
$4. x = 1 \wedge k = 0 \rightarrow k \geq 0 \wedge x = 2^k$	predicate logic
$5.\{T\}\ k \coloneqq 0; x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$	weaken postcondition 3,4

   - We can create a full proof outline following the proof: there is assignment, assignment, sequence and weaken postcondition; so, we can create:
     $\{T\}\ k \coloneqq 0; \{k = 0\}\ x \coloneqq 1\ \{x = 1 \wedge k = 0\} \{k \geq 0 \wedge x = 2^k\}$

2. Create a formal proof for the following full proof outline:
   $$\{T\}\{0 \geq 0 \wedge 1 = 2^0\}\ k \coloneqq 0; \{k \geq 0 \wedge 1 = 2^k\}\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$$
   - There are two statements, and they are in sequence.

$1.\{k \geq 0 \wedge 1 = 2^k\}\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$	backward assignment
$2.\{0 \geq 0 \wedge 1 = 2^0\}\ k \coloneqq 0\ \{k \geq 0 \wedge 1 = 2^k\}$	backward assignment
$3.\{0 \geq 0 \wedge 1 = 2^0\}\ k \coloneqq 0;\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$	sequence 2,1
$4. T \rightarrow 0 \geq 0 \wedge 1 = 2^0$	predicate logic
$5.\{T\}\ k \coloneqq 0;\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$	strengthen precondition 4,3

- Both Examples 5 and 6 are proving the triple $\{T\}\ k \coloneqq 0;\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$, but we have different proofs and different proof outlines. The reason here is that the forward assignment and the backward assignment are equally powerful.
- In general, a program can be proved in different ways. For each formal proof, there is a corresponding full proof outline.

<u>Minimal Proof Outlines and Partial Proof Outlines</u>

- In the previous class, we learned how to get a full proof outline from a formal proof: in a program, we wrap each statement with the triple used in the formal proof for this program.
  - For the sake of completeness, we don't have a good way to show conjunction/disjunction rule in a proof outline, since we need to wrap the same statement in different triples.

- In a **minimal proof outline**, we have the minimum amount of program annotation that allows us to infer the rest of the formal proof outline: in general, we can't infer the *initial precondition* and *initial postcondition*, nor can we infer the *invariants of loops*, so a minimal outline will include those conditions and no others.
- A **partial proof outline** is somewhere in the middle: More filled-in than a minimal outline but not completely full.

1. In Lecture 15, we have seen the following full proof outline:

$\{n \geq 0\}$
$k \coloneqq 0; \{n \geq 0 \wedge k = 0\}\ s \coloneqq 0;\ \{n \geq 0 \wedge k = 0 \wedge s = 0\}$
$\{\textbf{inv}\ p_1 \equiv 0 \leq k \leq n \wedge s = sum(0, k)\}$
while $k < n$ **do**
    $\{p_1 \wedge\ k < n\}$

$\{p_1[k+1\ /\ k][s+k+1\ /\ s]\}\ s \coloneqq s+k+1;\{p_1[k+1\ /\ k]\}\ k \coloneqq k+1\ \{p_1\}$

od

$\{p_1 \wedge k \geq n\}$

$\{s = sum\ (0,n)\}$

What is the minimal proof outline of the above statement? (Use red color)

2. Consider the following full proof outlines and their minimal proof outlines:
   a. $\{T\}\ k \coloneqq 0;\ \{k=0\}\ x \coloneqq 1\ \{x=1 \wedge k=0\}\{k \geq 0 \wedge x = 2^k\}$
   b. $\{T\}\{0 \geq 0 \wedge 1 = 2^0\}\ k \coloneqq 0;\{k \geq 0 \wedge 1 = 2^k\}\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$
   c. $\{T\}\ k \coloneqq 0;\ \{k=0\}\ \{k \geq 0 \wedge 1 = 2^k\}\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$

   - These full proof outlines have the same minimal proof outlines: $\{T\}\ k \coloneqq 0\ x \coloneqq 1\{k \geq 0 \wedge x = 2^k\}$. The reason multiple full proof outlines can have the same minimal outline is because different organizations of backward assignment and forward assignment can have the same minimal outline. There can also be differences in whether and where preconditions are strengthened or postconditions are weakened.

<u>Expanding Minimal/Partial Proof Outlines to Full Proof Outlines</u>

- Remember that, in a full proof outline, every statement needs to be wrapped in provable triple. To expand a minimal/partial proof outline into a full proof outline, basically we need to infer all the missing predicates. Postconditions are inferred from preconditions using $sp(\dots)$, and preconditions are inferred from postconditions using $wlp\ (\dots)$. Loop invariants tell us how to annotate the loop body and postcondition.

- Expanding a minimal/partial outline can lead to several different full outlines (because multiple full outlines can have the same minimal outline), but all the full outlines will be correct, and the differences between them are generally stylistic.

3. Expand the following minimal proof outline to a full proof outline.

$$\{y = x\}\ \textbf{if}\ x < 0\ \textbf{then}\ y \coloneqq -x\ \textbf{fi}\ \{y = abs\ (x)\}$$

   a. First, let's use wlp on each branch then use Conditional Rule 1 for the whole conditional statement. Then we have:

$\{y = x\}$
if $x < 0$ **then**
$\quad\{y = x \wedge x < 0\}\ \{-x = abs\ (x)\}\ y \coloneqq -x\ \{y = abs\ (x)\}$
else
$\quad\{y = x \wedge x \geq 0\}\ \{y = abs\ (x)\}$ **skip** $\{y = abs\ (x)\}$
fi $\{y = abs\ (x)\}$

   b. Let's use sp on each branch then use Conditional Rule 1 for the whole conditional statement. Then we have:

$\{y = x\}$
if $x < 0$ **then**
$\quad\{y = x \wedge x < 0\}\ y \coloneqq -x\ \{y_0 = x \wedge x < 0 \wedge y = -x\}\ \{y = abs\ (x)\}$
else
$\quad\{y = x \wedge x \geq 0\}$ **skip** $\{y = x \wedge x \geq 0\}\ \{y = abs\ (x)\}$
fi $\{y = abs\ (x)\}$

c. Let's use wlp on each branch then use Conditional Rule 2 for the whole conditional statement. Then we have:

$\{y = x\}$
$\{(x < 0 \to -x = abs\ (x)) \wedge (x \geq 0 \to y = abs\ (x))\}$
if $x < 0$ **then**
    $\{-x = abs\ (x)\}\ y \coloneqq -x\ \{y = abs\ (x)\}$
else
    $\{y = abs\ (x)\}$ **skip** $\{y = abs\ (x)\}$
fi $\{y = abs\ (x)\}$

4. Expand the following minimal proof outline to a full proof outline that's different from Example 1.
   - Let's use backward assignments before the loop and forward assignments in the loop body.

$\{n \geq 0\}$
$\{0 \leq 0 \leq n \wedge 0 = sum(0,0)\}$
$k \coloneqq 0; \{0 \leq k \leq n \wedge 0 = sum(0,k)\}\ s \coloneqq 0;$
$\{\mathbf{inv}\ p_1 \equiv 0 \leq k \leq n \wedge s = sum(0\ ,k)\}$
while $k < n$ **do**
    $\{p_1 \wedge\ k < n\}$
    $s \coloneqq s + k + 1; \{0 \leq k \leq n \wedge s_0 = sum(0\ ,k) \wedge\ k < n \wedge s = s_0 + k + 1\}$
    $k \coloneqq k + 1\ \{0 \leq k_0 \leq n \wedge s_0 = sum(0\ ,k_0) \wedge\ k_0 < n \wedge s = s_0 + k_0 + 1 \wedge k = k_0 + 1\}$
    $\{p_1\}$
od
$\{p_1 \wedge k \geq n\}$
$\{s = sum\ (0,n)\}$

Proofs for Total Correctness

- So far, all the proofs (formal proofs, full proof outlines, minimal and partial proof outlines) and all the inference rules we learned are for partial correctness (although most of them work for both correctness). Each proof system we have seen is a **proof system for partial correctness**, which is the set of logical formulas determined by the sets of axioms and rules of inference for partial correctness.
  - In other words, the triples $\{p\}\ S\ \{q\}$ we proved is provable under partial correctness: $\vdash \{p\}\ S\ \{q\}$.

- Now let's see how to come up with a **proof system for total correctness** which can prove a triple $\{p\}\ S\ \{q\}$ provable under total correctness: $\vdash_{tot} \{p\}\ S\ \{q\}$: we need to avoid possible runtime errors and divergence.

Inference Rules for Loop-free Statement under Total Correctness

- Previously, we have learned how to get wp from wlp: we include domain predicates for expressions $D(e)$, predicates $D(p)$, and statements $D(S)$ that guarantee the evaluation of e, p and S won't cause a runtime error.
- We use a similar idea to get $\vdash_{tot} \{p'\}\ S\ \{q'\}$ from $\vdash \{p\}\ S\ \{q\}$ by including domain predicates:
  - In general, for a loop-free S:
    - if we get $\vdash \{wlp(S,q)\}\ S\ \{q\}$, then for total correctness we have $\vdash_{tot} \{wp(S, q \wedge D(q))\}\ S\{q \wedge D(q)\}$.
    - if we get $\vdash \{p\}\ S\ \{sp(p,S)\}$, then for total correctness we have $\vdash_{tot} \{p \wedge D(p) \wedge D(S)\}\ S\ \{sp(p \wedge D(p) \wedge D(S), S)\}$.

- For a predicate p, we define the **safe version** of p as $\downarrow p \equiv p \wedge D(p)$.
  - With this definition we can rewrite $wp(S,q) \Leftrightarrow D(S) \wedge \downarrow wlp(S,q)$

- If a predicate $p \Leftrightarrow \downarrow p$, then we say p **is safe**.
- In a triple $\{p\}\ S\ \{q\}$, if p and q are both safe, then $\{p\}\ S\ \{q\}$ is **safe**. **In a proof of total correctness, we only prove safe triples.**
- If we are asked to prove a triple with unsafe precondition or postcondition under total correctness, it's not doable. All we can do is to add domain predicates to make the triple safe first. In general, for triple $\{p\}\ S\ \{q\}$, its **safe version** is $\{\downarrow p \wedge D(S)\}\ S\ \{\downarrow q\}$.