Prove that addition of natural numbers is associative.

```isar
primrec add :: "nat ⇒ nat ⇒ nat" where
  "add 0 n = n" |
  "add (Suc m) n = Suc (add m n)"
```

The first solution proceeds by induction on x.

```isar
theorem add_assoc: "add x (add y z) = add (add x y) z"
proof (induct x)
  show "add 0 (add y z) = add (add 0 y) z" by simp
next
  fix k assume IH: "add k (add y z) = add (add k y) z"
  have "add (Suc k) (add y z) = Suc (add k (add y z))" by simp
  also from IH have "... = Suc (add (add k y) z)" by simp
  also have "... = add (Suc (add k y)) z" by simp
  finally show "add (Suc k) (add y z) = add (add (Suc k) y) z" by simp
qed
```

Next we will proceed by induction on y. This proof requires two lemmas, both of which are also proved by induction.

```isar
lemma add_zero: "add x 0 = x"
proof (induct x)
  show "add 0 0 = 0" by simp
next
  fix k assume IH: "add k 0 = k"
  have "add (Suc k) 0 = Suc (add k 0)" by simp
  also from IH have "... = Suc k" by simp
  finally show "add (Suc k) 0 = Suc k".
qed
```

```isar
lemma add_one: "add x (Suc y) = Suc (add x y)"
proof (induct x)
  show "add 0 (Suc y) = Suc (add 0 y)" by simp
next
  fix x assume IH: "add x (Suc y) = Suc (add x y)"
  have "add (Suc x) (Suc y) = Suc (add x (Suc y))" by simp
  also from IH have "... = Suc (Suc (add x y))" by simp
```
also have ". . . = Suc (add (Suc x) y)" by simp
finally show "add (Suc x) (Suc y) = Suc (add (Suc x) y)" .
qed

theorem add_assoc_v2: "add x (add y z) = add (add x y) z"
proof (induct y)
  show "add x (add 0 z) = add (add x 0) z" using add_zero by simp
next
  fix k assume IH: "add x (add k z) = add (add x k) z"
  have "add x (add (Suc k) z) = add x (Suc (add k z))" by simp
also have ". . . = Suc (add x (add k z))" using add_one by simp
also from IH have ". . . = Suc (add (add x k) z)" by simp
also have ". . . = add (Suc (add x k)) z" by simp
also have ". . . = add (add x (Suc k)) z" using add_one by simp
finally show "add x (add (Suc k) z) = add (add x (Suc k)) z" .
qed

Last we proceed by induction on z. This proof requires the same two lemmas.

theorem add_assoc_v3: "add x (add y z) = add (add x y) z"
proof (induct z)
  show "add x (add y 0) = add (add x y) 0" using add_zero by simp
next
  fix k assume IH: "add x (add y k) = add (add x y) k"
  have "add x (add y (Suc k)) = add x (Suc (add y k))" using add_one by simp
also have ". . . = Suc (add x (add y k))" using add_one by simp
also from IH have ". . . = Suc (add (add x y) k)" by simp
also have ". . . = add (add x y) (Suc k)" using add_one by simp
finally show "add x (add y (Suc k)) = add (add x y) (Suc k)" .
qed