1.

Theorem plus_n_Sm : ∀ n m : nat,

S (n + m) = n + (S m).
Proof.
(* FILL IN HERE *) Admitted.

[ans]:

```
Theorem plus_n_Sm:forall n m:nat,
  S(n+m)=n+(S m).
Proof.
  intros n m. induction n as [|n' IHn'].
  - simpl. reflexivity.
  - simpl. rewrite->IHn'. reflexivity.
Qed.
```

[运行结果] :

```
Theorem plus_n_Sm:forall n m:nat,
  S(n+m)=n+(S m).
Proof.
  intros n m. induction n as [|n' IHn'].
  - simpl. reflexivity.
  - simpl. rewrite->IHn'. reflexivity.
Qed.
```

2.

Theorem add_shuffle3 : ∀ n m p : nat,
n + (m + p) = m + (n + p).
Proof.
(* FILL IN HERE *) Admitted.

[ans]:

```
Theorem add_shuffle3: forall n m p:nat,
  n+(m+p)=m+(n+p).
Proof.
  intros n m p. induction n as [|n' IHn'].
  - simpl. reflexivity.
  - simpl. rewrite->IHn'. rewrite->plus_n_Sm. reflexivity.
Qed.
```

[运行结果]：

```
Theorem add_shuffle3: forall n m p:nat,
  n+(m+p)=m+(n+p).
Proof.
  intros n m p. induction n as [|n' IHn'].
  - simpl. reflexivity.
  - simpl. rewrite->IHn'. rewrite->plus_n_Sm. reflexivity.
Qed.
```

3.

Theorem mul_n_Sm : forall n m : nat, n + n * m = n * S m.

Proof.
 (* FILL IN HERE *) Admitted.

[ans]:

```
Theorem mul_n_Sm:forall n m:nat,
  n+n*m=n*S m.
Proof.
  intros n m. induction n as [|n' IHn'].
  - simpl. reflexivity.
  - simpl. rewrite->add_shuffle3. rewrite->IHn'. reflexivity.
Qed.
```

[运行结果]：

```
Theorem mul_n_Sm:forall n m:nat,
  n+n*m=n*S m.
Proof.
  intros n m. induction n as [|n' IHn'].
  - simpl. reflexivity.
  - simpl. rewrite->add_shuffle3. rewrite->IHn'. reflexivity.
Qed.
```

4.

Theorem mul_comm : ∀ m n : nat,
 m × n = n × m.
Proof.
 (* FILL IN HERE *) Admitted.

[ans]:

```
Theorem mul_0_r : forall n:nat,
  n * 0 = 0.
Proof.
  intros n. induction n as [| n' IHn'].
  - reflexivity.
  - simpl. rewrite -> IHn'. reflexivity.
Qed.


Theorem mul_comm:forall n m:nat,
  m*n=n*m.
Proof.
  intros n m. induction n as [|n' IHn'].
  - simpl. rewrite->mul_0_r. reflexivity.
  - simpl. rewrite<-IHn'. rewrite->mul_n_Sm. reflexivity.
Qed.
```

[运行结果] :

```
Theorem mul_n_Sm:forall n m:nat,
  n+n*m=n*S m.
Proof.
  intros n m. induction n as [|n' IHn'].
  - simpl. reflexivity.
  - simpl. rewrite->add_shuffle3. rewrite->IHn'. reflexivity.
Qed.


Theorem mul_0_r : forall n:nat,
  n * 0 = 0.
Proof.
  intros n. induction n as [| n' IHn'].
  - reflexivity.
  - simpl. rewrite -> IHn'. reflexivity.
Qed.


Theorem mul_comm:forall n m:nat,
  m*n=n*m.
Proof.
  intros n m. induction n as [|n' IHn'].
  - simpl. rewrite->mul_0_r. reflexivity.
  - simpl. rewrite<-IHn'. rewrite->mul_n_Sm. reflexivity.
Qed.
```