Exercise 2

The following algebraic specification of natural numbers is given:

```
sort Nat;
     operations
          zero : -> Nat;
          succ : Nat -> Nat;
          add : Nat x Nat -> Nat;
     declare i, j : Z;
     axioms
          add(zero,i) == i; /* A1*/
          add(succ(i),j) == succ(add(i,j)); /* A2 */
          i == add(i,zero); /* A3, proved in lecture */
```

Prove the following theorem:

```
          add(i,j) == add(j,i);
```

Exercise 3

Write an algebraic specification for queues of natural numbers. Queues should offer the following operations:

- `newqueue`: creates a new, i.e., empty queue
- `isempty`: checks whether the queue is empty
- `enqueue`: adds a number to the tail of the queue
- `dequeue`: removes a number from the head of the queue
- `head`: returns the number at the head of the queue (i.e., the next number to be dequeued)
- `tail`: returns the number at the tail of the queue (i.e., the last number which was enqueued)

Add axioms for defining the semantics of these operations. In particular, take care that you express the First-In-First-Out (FIFO) property of queues: The first number which is enqueued will also be the first one to be dequeued.