1.

分析：

根据提示，Fixpoint中的定义必须都是递归的，所以这里需要同时对l1和l2进行模式匹配。基线条件就是l1或l2中有一个是空，那么就返回另外一个。若都非空则将l1和l2的头元素重新组合成列表再在后面进行递归。

代码：

```
(** **** Exercise: 3 stars, advanced (alternate)

    Complete the following definition of [alternate], which
    interleaves two lists into one, alternating between elements taken
    from the first list and elements from the second.  See the tests
    below for more specific examples.

    Hint: there is an elegant way of writing [alternate] that fails to
    satisfy Coq's requirement that all [Fixpoint] definitions be
    _structurally recursing_, as mentioned in [Basics]. If you
    encounter that difficulty, consider pattern matching against both
    lists at the same time with the "multiple pattern" syntax we've
    seen before. *)

Fixpoint alternate (l1 l2 : natlist) : natlist :=
  match l1, l2 with
  | nil, _ => l2
  | _, nil => l1
  | h1 :: t1, h2 :: t2 => h1 :: h2 :: (alternate t1 t2)
  end.


Example test_alternate1:
  alternate [1;2;3] [4;5;6] = [1;4;2;5;3;6].
Proof. reflexivity.
  (* FILL IN HERE *) Admitted.

Example test_alternate2:
  alternate [1] [4;5;6] = [1;4;5;6].
Proof. reflexivity.
  (* FILL IN HERE *) Admitted.

Example test_alternate3:
  alternate [1;2;3] [4] = [1;4;2;3].
Proof. reflexivity.
  (* FILL IN HERE *) Admitted.
```

运行结果：

```
Fixpoint alternate (l1 l2 : natlist) : natlist :=
  match l1, l2 with
  | nil, _ => l2
  | _, nil => l1
  | h1 :: t1, h2 :: t2 => h1 :: h2 :: (alternate t1 t2)
  end.

Example test_alternate1:
  alternate [1;2;3] [4;5;6] = [1;4;2;5;3;6].
Proof. reflexivity.
  (* FILL IN HERE *) Admitted.

Example test_alternate2:
  alternate [1] [4;5;6] = [1;4;5;6].
Proof. reflexivity.
  (* FILL IN HERE *) Admitted.

Example test_alternate3:
  alternate [1;2;3] [4] = [1;4;2;3].
Proof. reflexivity.
  (* FILL IN HERE *) Admitted.
```

2.

分析：

由于不知道题目中所说的集合中是否包含重复的元素，所以两种理解都进行了实现。实现的方法都是递归。

若不能出现相同元素，则通过已经实现的count函数，判断其返回值是否为0，就能够判断该元素是否在交集之中。

若能出现相同元素，则需要在递归函数中用已经实现的remove_one函数进行处理。

代码：

```
Fixpoint inter (s1: bag) (s2: bag) : bag :=
  match s1 with
  | nil => nil
  | h :: t => if ((count h s2) =? 0) then inter t s2
               else h :: inter t s2
  end.

Example test_inter1: inter [1;3;2;4;5] [1;2;3] = [1;3;2].
Proof. simpl. reflexivity. Qed.

Example test_inter2: inter [1;3;2;4;5] [ ] = [ ].
Proof. simpl. reflexivity. Qed.

Example test_inter3: inter [] [1;2;3] = [].
Proof. simpl. reflexivity. Qed.

```

```
Example test_inter4: inter [1;3;5] [1;2;3;5;7;9] = [1;3;5].
Proof. simpl. reflexivity. Qed.


```

```
Fixpoint inter' (s1: bag) (s2: bag) : bag :=
  match s1, s2 with
  | nil, _ => nil
  | _, nil => nil
  | h1 :: t1, _ => if ((count h1 s2) =? 0) then inter t1 s2
                       else h1 :: inter' t1 (remove_one h1 s2)
  end.


Example test_inter'1: inter' [1;3;1;1;2;4;5] [1;2;1;3] = [1;3;1;2].
Proof. simpl. reflexivity. Qed.

Example test_inter'2: inter' [1;3;3;2;4;5] [ ] = [ ].
Proof. simpl. reflexivity. Qed.

Example test_inter'3: inter' [] [1;2;3;1] = [].
Proof. simpl. reflexivity. Qed.

Example test_inter'4: inter' [1;3;5;8;6;1] [1;2;3;5;7;1;9] = [1;3;5;1].
Proof. simpl. reflexivity. Qed.
```

运行结果：

```
Fixpoint inter (s1: bag) (s2: bag) : bag :=
  match s1 with
  | nil => nil
  | h :: t => if ((count h s2) =? 0) then inter t s2
             else h :: inter t s2
  end.

Example test_inter1: inter [1;3;2;4;5] [1;2;3] = [1;3;2].
Proof. simpl. reflexivity. Qed.

Example test_inter2: inter [1;3;2;4;5] [ ] = [ ].
Proof. simpl. reflexivity. Qed.

Example test_inter3: inter [] [1;2;3] = [].
Proof. simpl. reflexivity. Qed.

Example test_inter4: inter [1;3;5] [1;2;3;5;7;9] = [1;3;5].
Proof. simpl. reflexivity. Qed.
```

```
Fixpoint inter' (s1: bag) (s2: bag) : bag :=
  match s1, s2 with
  | nil, _ => nil
  | _, nil => nil
  | h1 :: t1, _ => if ((count h1 s2) =? 0) then inter t1 s2
                       else h1 :: inter' t1 (remove_one h1 s2)
  end.


Example test_inter'1: inter' [1;3;1;1;2;4;5] [1;2;1;3] = [1;3;1;2].
Proof. simpl. reflexivity. Qed.

Example test_inter'2: inter' [1;3;3;2;4;5] [ ] = [ ].
Proof. simpl. reflexivity. Qed.

Example test_inter'3: inter' [] [1;2;3;1] = [].
Proof. simpl. reflexivity. Qed.

Example test_inter'4: inter' [1;3;5;8;6;1] [1;2;3;5;7;1;9] = [1;3;5;1].
Proof. simpl. reflexivity. Qed.
```