1 prove that b is correct answer for argument nil

2 prove that e(f xs)) is correct ans. for arg. (x::xs)

```
CIS505/705 Exercise   Sample solution

Here is the list-induction rule:

                              +-----------------------------------
                              | (x::xs)  : 'a list        assumption
        b : P(nil, b)         |  (f xs) : P(xs, (f xs))   assumption
                              |    ...
                              |  e(f xs)  : P(x::xs, e(f xs))
                              +-----------------------------------
(list ind)---------------------------------------------------------
          fun f  nil      =  b
          |    f (x::xs)  =  e(f xs)
                                : Forall arg Exists ans : P(arg, ans)

Here is a correctness property:

SUM(ms, n) ==    (i) ms = [n0, n1,..., nk]
                 and
                 (ii) n = 0 + n0 + n1 + ... + nk

Finish the proof that function  sumit  has the  SUM  correctness property:

 |-  fun sumit nil = 0
     |   sumit (x::xs) = x + (sumit xs)
                                  : Forall arg Exists ans : SUM(arg, ans)

1. 0 : SUM([], 0)             because (i)  [] has no ints in it,
                                      (ii)  0 is the sum of zero ints.
                              So,  property  SUM([], 0) holds true.
                              So,  0 has "data type"  SUM([], 0)
+---------------------------------------
|
| 2.   (x::xs) : int list               assumption --- say the list is nonempty


| 3.   (sumit xs) : SUM(xs, sumit xs)   assumption --- say the recursive call,
                                               (sumit xs),  returns
                                               the correct answer for arg  xs

| 4.   x + (sumit xs)  : SUM(x::xs, x+(sumit xs))

                              because, Line 3 says:
                                (i) xs = [n0, n1,..., nk],
                                (ii) (sumit xs) = 0 + n0 + n1 + ... + nk

                              So, we have that
                                (i) x::xs = [x, n0, n1,..., nk]
                                (ii) x + (sumit xs) = x + 0 + n0 + n1 + ... + nk
                                                    = 0 + x + n0 + n1 + ... + nk
                              So,  SUM(x::xs, x+(sumit xs))  holds true.
                              So,  x + (sumit xs)  has "data type"
                                                 SUM(x::xs, x+(sumit xs))
+-----------------------------------------
5.  fun sumit nil = 0
    |    sumit (x::xs) = x + (sumit xs)
                             : Forall arg Exists ans : SUM(arg, ans)

                                                  by list ind 1, 2-4
```