```
foldl f z [] = [z]
foldl f z (x:xs) = foldl f (f z x) xs
flip f x y = f y x
reverse = foldl (flip (:)) []
palin xs = reverse xs == xs
Ex1.hs> :debug palin
reverse :: [a] -> [[a]]
Ex1.hs> is this type correct> n
flip :: (a -> b -> c) -> b -> a -> c
Ex1.hs> is this type correct> y
foldl :: (a -> b -> a) -> a -> [b] -> [a]
Ex1.hs> is this type correct> n
type error - contributing locations
foldl f z [] = [z]
foldl f z (x:xs) = foldl f (f z x) xs
```

Chameleon Interactive Debugging

```
*Main> :load split.hs
split.hs:1: ERROR: Type error
Problem: Case alternatives incompatible
Types : [a] -> (b, a)
         [a] -> (c, [a])
Conflict: split xs = case xs of
                       [] -> ([], [])
                      [x] -> ([], x)
                       (x:y:zs) \rightarrow let (xs, ys) = split zs
                                   in (x:xs, y:ys)
*Main>
```

Chameleon Type Error