A lifetime cannot be determined in the given situation.

Erroneous code example:

```
fn transmute_lifetime<'a, 'b, T>(t: &'a (T,)) -> &'b T {
    match (&t,) { // error!
        ((u,),) => u,
    }
}

let y = Box::new((42,));
let x = transmute_lifetime(&y);
```

In this code, you have two ways to solve this issue: 1. Enforce that `'a` lives at least as long as `'b`. 2. Use the same lifetime requirement for both input and output values.

So for the first solution, you can do it by replacing `'a` with `'a: 'b`:

```
fn transmute_lifetime<'a: 'b, 'b, T>(t: &'a (T,)) -> &'b T {
    match (&t,) { // ok!
        ((u,),) => u,
    }
}
```

In the second you can do it by simply removing `'b` so they both use `'a`:

```
fn transmute_lifetime<'a, T>(t: &'a (T,)) -> &'a T {
    match (&t,) { // ok!
        ((u,),) => u,
    }
}
```