

Objective Caml version 3.06

```
# type rat = int * int;;
type rat = int * int

# let inverse ((a,b): rat) = (b,a);;
val inverse : rat -> int * int = <fun>

# let inverse ((a,b): rat): rat = (b,a);;
val inverse : rat -> rat = <fun>

# let mul ((a1,b1), (a2,b2)): rat * rat): rat = (a1*a2,b1*b2);;
val mul : rat * rat -> rat = <fun>

# mul ((1,2), (1,4));;
- : rat = (1, 8)

# let add ((a1,b1), (a2,b2)): rat * rat): rat = (a1*b2 + a2*b1, b1*b2);;
val add : rat * rat -> rat = <fun>

# add ((1,2), (1,4));;
- : rat = (6, 8)

# let mul ((a1,b1): rat) ((a2,b2): rat): rat = (a1*a2,b1*b2);;
val mul : rat -> rat -> rat = <fun>

# mul (1,2) (1,4);;
- : rat = (1, 8)

# let half = mul (1,2);;
val half : rat -> rat = <fun>

# half (3,4);;
- : rat = (3, 8)

#
```