

Objective Caml version 3.06

```
# let fact = fun x -> if x = 0 then 1 else x * fact (x - 1);;
Characters 45-49:
  let fact = fun x -> if x = 0 then 1 else x * fact (x - 1);;
                                     ^^^^^
```

Unbound value fact

```
# let rec fact = fun x -> if x = 0 then 1 else x * fact (x - 1);;
val fact : int -> int = <fun>
```

```
# fact 6;;
- : int = 720
```

```
# let rec fact x = if x = 0 then 1 else x * fact (x - 1);;
val fact : int -> int = <fun>
```

```
# let exp = fun (x, n) -> if n = 0 then 1 else x * exp (x, n-1);;
Characters 54-60:
  let exp = fun (x, n) -> if n = 0 then 1 else x * exp (x, n-1);;
                                     ^^^^^^^
```

*This expression has type int * int but is here used with type float*

```
# exp;;
- : float -> float = <fun>
```

```
# let rec exp = fun (x, n) -> if n = 0 then 1 else x * exp (x, n-1);;
val exp : int * int -> int = <fun>
```

```
# exp;;
- : int * int -> int = <fun>
```

```
# let exp (x, n) = if n = 0 then 1 else x * exp (x, n - 1);;
val exp : int * int -> int = <fun>
```

```
# exp (2, 10);;
- : int = 1024
```

```
# let exp_curry x n = if n = 0 then 1 else x * exp (x, n - 1);;
val exp_curry : int -> int -> int = <fun>
```

```
# exp_curry (2, 10);;
Characters 11-16:
  exp_curry (2, 10);;
             ^^^^^
```

*This expression has type int * int but is here used with type int*

```
# exp_curry 2 10;;
- : int = 1024
```

```
# exp_curry 2;;
- : int -> int = <fun>
```

```
# let exp2 = exp_curry 2;;
val exp2 : int -> int = <fun>
```

```
# exp2 10;;
- : int = 1024
```

```
#
```