```
Last login: Mon Jan 23 13:15:05 on ttys012
carbon:SamplePrograms$ utop
────────────────┬──────────────────────────────────────────────────────────────┬────────────
                │ Welcome to utop version 1.14 (using OCaml version 4.01.0)! │
                └──────────────────────────────────────────────────────────────┘

Type #utop_help for help about using utop.

─( 18:00:00 )─< command 0 >──────────────────────────────────────────{ counter: 0 }─
utop # let add x y = x + y ;;
val add : int -> int -> int = <fun>
─( 13:27:46 )─< command 1 >──────────────────────────────────────────{ counter: 0 }─
utop # let add' = fun x -> fun y -> x + y ;;
val add' : int -> int -> int = <fun>
─( 13:27:54 )─< command 2 >──────────────────────────────────────────{ counter: 0 }─
utop # add 3 4;;
- : int = 7
─( 13:28:49 )─< command 3 >──────────────────────────────────────────{ counter: 0 }─
utop # add' 3 4 ;;
- : int = 7
─( 13:28:53 )─< command 4 >──────────────────────────────────────────{ counter: 0 }─
utop # let inc = add 1 ;;
val inc : int -> int = <fun>
─( 13:28:56 )─< command 5 >──────────────────────────────────────────{ counter: 0 }─
utop # inc 4 ;;
- : int = 5
─( 13:29:54 )─< command 6 >──────────────────────────────────────────{ counter: 0 }─
utop # inc 6 ;;
- : int = 7
─( 13:30:00 )─< command 7 >──────────────────────────────────────────{ counter: 0 }─
utop # let inc' n = add 1 n ;;
val inc' : int -> int = <fun>
─( 13:30:03 )─< command 8 >──────────────────────────────────────────{ counter: 0 }─
utop # inc ' 7 ;;
Error: Syntax error
─( 13:38:26 )─< command 9 >──────────────────────────────────────────{ counter: 0 }─
utop # inc' 7 ;;
- : int = 8
─( 13:38:30 )─< command 10 >─────────────────────────────────────────{ counter: 0 }─
utop # let inc'' = fun n -> n + 1 ;;
val inc'' : int -> int = <fun>
─( 13:38:34 )─< command 11 >─────────────────────────────────────────{ counter: 0 }─
utop # let the_answer = 42 ;;
val the_answer : int = 42
─( 13:40:17 )─< command 12 >─────────────────────────────────────────{ counter: 0 }─
utop # let fact n = if n = 0 then 1 else n * (fact (n-1)) ;;
Error: Unbound value fact
─( 13:40:25 )─< command 13 >─────────────────────────────────────────{ counter: 0 }─
utop # let rec fact n = if n = 0 then 1 else n * (fact (n-1)) ;;
val fact : int -> int = <fun>
─( 13:49:32 )─< command 14 >─────────────────────────────────────────{ counter: 0 }─
utop # let rec fact' n = if n < 0 then -100
                         else if n = 0 then 1
```

```
                                else n * (fact (n-1)) ;;
val fact' : int -> int = <fun>
─( 13:49:49 )─< command 15 >──────────────────────────────────────{ counter: 0 }─
utop # fact' 4 ;;
- : int = 24
─( 13:50:58 )─< command 16 >──────────────────────────────────────{ counter: 0 }─
utop # fact' -2 ;
;;
Error: This expression has type int -> int
       but an expression was expected of type int
─( 13:51:01 )─< command 17 >──────────────────────────────────────{ counter: 0 }─
utop # fact' (-2) ;;
- : int = -100
─( 13:51:08 )─< command 18 >──────────────────────────────────────{ counter: 0 }─
utop # let rec fact n = if n = 0 then 1 else n * fact (n-1) ;;
val fact : int -> int = <fun>
─( 13:51:17 )─< command 19 >──────────────────────────────────────{ counter: 0 }─
utop # let rec fact n = if n = 0 then 1 else n * fact n-1 ;;
val fact : int -> int = <fun>
─( 13:52:23 )─< command 20 >──────────────────────────────────────{ counter: 0 }─
utop # fact 4 ;;
Stack overflow during evaluation (looping recursion?).
─( 13:52:48 )─< command 21 >──────────────────────────────────────{ counter: 0 }─
utop # (-) ;;
- : int -> int -> int = <fun>
─( 13:52:56 )─< command 22 >──────────────────────────────────────{ counter: 0 }─
utop # fact (-) ;;
Error: This expression has type int -> int -> int
       but an expression was expected of type int
─( 13:54:12 )─< command 23 >──────────────────────────────────────{ counter: 0 }─
utop # fact (~-) ;;
Error: This expression has type int -> int
       but an expression was expected of type int
─( 13:54:23 )─< command 24 >──────────────────────────────────────{ counter: 0 }─
utop # let cube x = x *. x *. x ;;
val cube : float -> float = <fun>
─( 13:54:42 )─< command 25 >──────────────────────────────────────{ counter: 0 }─
utop # let power n x = if n = 0 then 1.0 else x *. power (n-1) x ;;
Error: Unbound value power
─( 13:59:26 )─< command 26 >──────────────────────────────────────{ counter: 0 }─
utop # let rec power n x = if n = 0 then 1.0 else x *. power (n-1) x ;;
val power : int -> float -> float = <fun>
─( 13:59:45 )─< command 27 >──────────────────────────────────────{ counter: 0 }─
utop # power 3 3.4 ;;
- : float = 39.3039999999999949
─( 13:59:49 )─< command 28 >──────────────────────────────────────{ counter: 0 }─
utop # power 3 3.0 ;;
- : float = 27.
─( 13:59:53 )─< command 29 >──────────────────────────────────────{ counter: 0 }─
utop # let cube x = power 3 x ;;
val cube : float -> float = <fun>
─( 14:00:07 )─< command 30 >──────────────────────────────────────{ counter: 0 }─
utop # cube 3.4 ;;
```

```
- : float = 39.3039999999999949
─( 14:00:51 )─< command 31 >──────────────────────────────────────{ counter: 0 }─
utop # let cube = power 3;;
val cube : float -> float = <fun>
─( 14:00:55 )─< command 32 >──────────────────────────────────────{ counter: 0 }─
utop # let xs = 1 :: 2 :: 3 :: [] ;;
val xs : int list = [1; 2; 3]
─( 14:01:17 )─< command 33 >──────────────────────────────────────{ counter: 0 }─
utop # let xs' = [1;2;3] ;;
val xs' : int list = [1; 2; 3]
─( 14:09:34 )─< command 34 >──────────────────────────────────────{ counter: 0 }─
utop # xs = xs' ;;
- : bool = true
─( 14:10:04 )─< command 35 >──────────────────────────────────────{ counter: 0 }─
utop # let is_empty xs =
           match xs with
           | [] -> true
           | _  => false
;;
Error: Syntax error
─( 14:10:09 )─< command 36 >──────────────────────────────────────{ counter: 0 }─
utop # let is_empty xs =
           match xs with
           | [] -> true
           | _  -> false
;;
val is_empty : 'a list -> bool = <fun>
─( 14:13:45 )─< command 37 >──────────────────────────────────────{ counter: 0 }─
utop # is_empty [] ;;
- : bool = true
─( 14:13:51 )─< command 38 >──────────────────────────────────────{ counter: 0 }─
utop # (1, "hello");;
- : int * string = (1, "hello")
─( 14:13:56 )─< command 39 >──────────────────────────────────────{ counter: 0 }─
utop # let add x y = x + y ;;
val add : int -> int -> int = <fun>
─( 14:14:17 )─< command 40 >──────────────────────────────────────{ counter: 0 }─
utop # let add' (x,y) = x + y ;;
─( 14:15:04 )─< command 41 >──────────────────────────────────────{ counter: 0 }─
utop #
```