```
Last login: Fri Apr 14 13:49:43 on ttys009
carbon:SamplePrograms$ cd Intervals/v4
carbon:v4$ ls
intInterval.ml          stringInterval.ml
intervals.ml            useInterval.ml
carbon:v4$ ocamlbuild useInterval.byte
Finished, 9 targets (0 cached) in 00:00:00.
carbon:v4$ ./useInterval.byte
An interval: (3, 4)
Another interval: (3, 6)
Their intresection: (3, 4)
A string interval: (a, d)
carbon:v4$ cd ../v5
carbon:v5$ 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 # #mod_use "interval.ml";;
Cannot find file interval.ml.                                          —
( 14:08:33 )—< command 1 >————————————————————————{ counter: 0 }—ut
op # #mod_use "intervals.ml";;
module Intervals :
 sig
  module type Comparable =
      sig type t val compare : t -> t -> int val to_string : t -> string end
    module type Interval_intf =
      sig
        type t
        type endpoint
        val create : endpoint -> endpoint -> t
        val is_empty : t -> bool
        val contains : t -> endpoint -> bool
        val intersect : t -> t -> t
        val to_string : t -> string
      end
    module Make_interval : functor (Endpoint : Comparable) -> Interval_intf
  end
—( 14:08:44 )—< command 2 >————————————————————————{ counter: 0 }—
utop # #use "intInterval.ml";;
module Int_interval : Intervals.Interval_intf                           F
ile "intInterval.ml", line 27, characters 28-29:                       Er
ror: This expression has type int but an expression was expected of type
         Int_interval.endpoint
—( 14:09:06 )—< command 3 >————————————————————————{ counter: 0 }—
utop # #quit;;
carbon:v5$ cd ../v6                                                     c
arbon:v6$ 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 # #mod_use "intervals.ml";;
module Intervals :
 sig
  module type Comparable =
      sig type t val compare : t -> t -> int val to_string : t -> string end
    module type Interval_intf =
      sig
        type t
        type endpoint
        val create : endpoint -> endpoint -> t
        val is_empty : t -> bool
        val contains : t -> endpoint -> bool
        val intersect : t -> t -> t
        val to_string : t -> string
      end
    module Make_interval :
      functor (Endpoint : Comparable) ->
        sig
          type t
          type endpoint = Endpoint.t
          val create : endpoint -> endpoint -> t
          val is_empty : t -> bool
          val contains : t -> endpoint -> bool
          val intersect : t -> t -> t
          val to_string : t -> string
        end
  end
─( 14:14:09 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "intInterval.ml";;
module Int_comparable :
 sig type t = int val compare : t -> t -> t val to_string : t -> string end    mo
dule Int_interval :
  sig
    type t = Intervals.Make_interval(Int_comparable).t
    type endpoint = int
    val create : endpoint -> endpoint -> t
    val is_empty : t -> bool
    val contains : t -> endpoint -> bool
    val intersect : t -> t -> t
    val to_string : t -> string
  end
val i : Int_interval.t = <abstr>
─( 14:14:38 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop #
```

Arg	Arith_status	Array	ArrayLabels	Assert_failure	Big_int	Bigarray	Buffer	Callb