```
Last login: Wed Nov 26 11:08:04 on ttys011
carbon:code-examples$ utop
────────────────┬─────────────────────────────────────────────────────────────┬──────────
                │ Welcome to utop version 1.14 (using OCaml version 4.01.0)!  │
                └─────────────────────────────────────────────────────────────┘

Findlib has been successfully loaded. Additional directives:
  #require "package";;      to load a package
  #list;;                   to list the available packages
  #camlp4o;;                to load camlp4 (standard syntax)
  #camlp4r;;                to load camlp4 (revised syntax)
  #predicates "p,q,...";;   to set these predicates
  Topfind.reset();;         to force that packages will be reloaded
  #thread;;                 to enable threads


Type #utop_help for help about using utop.

─( 18:00:00 )─< command 0 >──────────────────────────────────────{ counter: 0 }─
utop # #use "wolf.ml";;
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val is_not_elem : 'a list -> 'a -> bool = <fun>
type loc = L | R
type state = loc * loc * loc * loc
val ok_state : state -> bool = <fun>
val final : loc * loc * loc * loc -> bool = <fun>
val other_side : loc -> loc = <fun>
val moves : state -> state list = <fun>
val crossing_v1 : unit -> state list option = <fun>
exception FoundPath of (loc * loc * loc * loc) list
File "wolf.ml", line 113, characters 9-12:
Error: Unbound value run
─( 11:26:45 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "wolf.ml";;
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val is_not_elem : 'a list -> 'a -> bool = <fun>
val run : 'a -> unit = <fun>
type loc = L | R
type state = loc * loc * loc * loc
val ok_state : state -> bool = <fun>
val final : loc * loc * loc * loc -> bool = <fun>
val other_side : loc -> loc = <fun>
val moves : state -> state list = <fun>
val crossing_v1 : unit -> state list option = <fun>
exception FoundPath of (loc * loc * loc * loc) list
val crossing_v2 : unit -> (loc * loc * loc * loc) list option = <fun>
exception KeepLooking
File "wolf.ml", line 137, characters 8-15:
Error: Unbound value is_elem
Did you mean is_some?
─( 11:26:52 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # #use "wolf.ml";;
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val is_not_elem : 'a list -> 'a -> bool = <fun>
val run : 'a -> unit = <fun>
val foldr : ('a -> 'b -> 'b) -> 'b -> 'a list -> 'b = <fun>
val is_elem : 'a -> 'a list -> bool = <fun>
type loc = L | R
type state = loc * loc * loc * loc
val ok_state : state -> bool = <fun>
val final : loc * loc * loc * loc -> bool = <fun>
val other_side : loc -> loc = <fun>
val moves : state -> state list = <fun>
val crossing_v1 : unit -> state list option = <fun>
exception FoundPath of (loc * loc * loc * loc) list
val crossing_v2 : unit -> (loc * loc * loc * loc) list option = <fun>
exception KeepLooking
val process_solution_exn : ('a -> string) -> 'a -> 'a option = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
val show_loc : loc -> string = <fun>
val show_state : loc * loc * loc * loc -> string = <fun>
val show_path : (loc * loc * loc * loc) list -> string = <fun>
val crossing_v3 : unit -> state list option = <fun>
─( 11:27:13 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # crossing_v1 () ;;
- : state list option =
Some
 [(L, L, L, L); (R, L, R, L); (L, L, R, L); (R, R, R, L); (L, R, L, L);
  (R, R, L, R); (L, R, L, R); (R, R, R, R)]
─( 11:27:41 )─< command 4 >──────────────────────────────────────{ counter: 0 }─
utop # crossing_v2 () ;;
- : (loc * loc * loc * loc) list option =
Some
 [(L, L, L, L); (R, L, R, L); (L, L, R, L); (R, R, R, L); (L, R, L, L);
  (R, R, L, R); (L, R, L, R); (R, R, R, R)]
─( 11:27:50 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # crossing_v3 () ;;
Here is a solution:
[ (L, L, L, L); (R, L, R, L); (L, L, R, L); (R, R, R, L); (L, R, L, L); (R, R, L
, R); (L, R, L, R); (R, R, R, R) ]
Do you like it?
n
Here is a solution:
[ (L, L, L, L); (R, L, R, L); (L, L, R, L); (R, L, R, R); (L, L, L, R); (R, R, L
, R); (L, R, L, R); (R, R, R, R) ]
Do you like it?
n
- : state list option = None
─( 11:27:58 )─< command 6 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;
carbon:code-examples$ utop
─────────┬────────────────────────────────────────────────────────────┬─────────
         │ Welcome to utop version 1.14 (using OCaml version 4.01.0)! │
         └────────────────────────────────────────────────────────────┘

Findlib has been successfully loaded. Additional directives:
  #require "package";;      to load a package
  #list;;                   to list the available packages
  #camlp4o;;                to load camlp4 (standard syntax)
  #camlp4r;;                to load camlp4 (revised syntax)
  #predicates "p,q,...";;   to set these predicates
  Topfind.reset();;         to force that packages will be reloaded
  #thread;;                 to enable threads


Type #utop_help for help about using utop.

─( 18:00:00 )─< command 0 >───────────────────{ counter: 0 }─
utop # #use "subsetsum_with_modules.ml";;
File "subsetsum_with_modules.ml", line 29, characters 8-23:
Error: Unbound module OurList
Did you mean List?
─( 11:37:48 )─< command 1 >───────────────────{ counter: 0 }─
utop # #mod_use "ourList.ml" ;;
module OurList :
  sig
    val map : ('a -> 'b) -> 'a list -> 'b list
    val filter : ('a -> bool) -> 'a list -> 'a list
    val foldr : ('a -> 'b -> 'b) -> 'b -> 'a list -> 'b
    val foldl : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a
    val is_elem : 'a -> 'a list -> bool
    val length : 'a list -> int
    val sum : int list -> int
    val to_string : ('a -> string) -> 'a list -> string
  end
─( 11:38:07 )─< command 2 >───────────────────{ counter: 0 }─
utop # foldr ;;
Error: Unbound value foldr
Did you mean floor?
─( 11:38:48 )─< command 3 >───────────────────{ counter: 0 }─
utop # OurList.foldr ;;
- : ('a -> 'b -> 'b) -> 'b -> 'a list -> 'b = <fun>
─( 11:39:07 )─< command 4 >───────────────────{ counter: 0 }─
utop # #use "subsetsum_with_modules.ml";;
val process_solution_option : ('a -> string) -> 'a -> 'a option = <fun>
val subsetsum_option : int list -> int list option = <fun>
val s : int list = [1; 3; -2; 5; -6]
Here is a solution:
[ 1; 5; -6 ]
Do you like it?
n
Here is a solution:
[ 3; -2; 5; -6 ]
Do you like it?
n
Oh no, no solution
─( 11:39:14 )─< command 5 >───────────────────{ counter: 0 }─
utop # #quit ;;
carbon:code-examples$ corebuild subsetsum_with_modules.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 subsetsum_with_modules.ml > subsetsum_with_modules.ml.depends
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 ourList.ml > ourList.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
```

```
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o ourList.cmo ourList.ml
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o subsetsum_with_modules.cmo subsetsum_with_modules.ml
ocamlfind ocamlc -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax -p
ackage sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -pac
kage core ourList.cmo subsetsum_with_modules.cmo -o subsetsum_with_modules.byte
carbon:code-examples$ ./subsetsum_with_modules.byte
Here is a solution:
[ 1; 5; -6 ]
Do you like it?
n
Here is a solution:
[ 3; -2; 5; -6 ]
Do you like it?
y
Thanks for playing...
Yeah, a solution.
[ 3; -2; 5; -6 ]
carbon:code-examples$ ./subsetsum_with_modules.native
-bash: ./subsetsum_with_modules.native: No such file or directory
carbon:code-examples$ corebuild subsetsum_with_modules.native
ocamlfind ocamlopt -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bi
n-annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package s
explib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package cor
e -o ourList.cmx ourList.ml
ocamlfind ocamlopt -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bi
n-annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package s
explib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package cor
e -o subsetsum_with_modules.cmx subsetsum_with_modules.ml
ocamlfind ocamlopt -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax
-package sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -p
ackage core ourList.cmx subsetsum_with_modules.cmx -o subsetsum_with_modules.nat
ive
carbon:code-examples$ ./subsetsum_with_modules.native
Here is a solution:
[ 1; 5; -6 ]
Do you like it?
n
Here is a solution:
[ 3; -2; 5; -6 ]
Do you like it?
y
Thanks for playing...
Yeah, a solution.
[ 3; -2; 5; -6 ]
carbon:code-examples$ ls _build/
_digests                                ourList.cmt
_log                                    ourList.cmx
lab_02.annot                            ourList.ml
lab_02.byte*                            ourList.ml.depends
lab_02.cmi                              ourList.o
lab_02.cmo                              subsetsum_with_modules.annot
```

```
lab_02.cmt                              subsetsum_with_modules.byte*
lab_02.cmti                             subsetsum_with_modules.cmi
lab_02.ml                               subsetsum_with_modules.cmo
lab_02.ml.depends                       subsetsum_with_modules.cmt
lab_02.mli                              subsetsum_with_modules.cmx
lab_02.mli.depends                      subsetsum_with_modules.ml
ocamlc.where                            subsetsum_with_modules.ml.depends
ourList.annot                           subsetsum_with_modules.native*
ourList.cmi                             subsetsum_with_modules.o
ourList.cmo
carbon:code-examples$ ls -l subsetsum_with_modules.byte
lrwxr-xr-x  1 evw  wheel  101 Nov 26 11:42 subsetsum_with_modules.byte@ -> /proj
ect/evw/Teaching/14_Fall_2041/public-class-repo/code-examples/_build/subsetsum_w
ith_modules.byte
carbon:code-examples$ cd Intervals/v1
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 useIntInterval.ml > useIntInterval.ml.depends
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intInterval.ml > intInterval.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intInterval.cmo intInterval.ml
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o useIntInterval.cmo useIntInterval.ml
ocamlfind ocamlc -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax -p
ackage sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -pac
kage core intInterval.cmo useIntInterval.cmo -o useIntInterval.byte
carbon:v1$ ./useIntInterval.byte
An interval: (3, 4)
Another interval: (3, 6)
Their intresection: (3, 4)
carbon:v1$
carbon:v1$
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 useIntInterval.ml > useIntInterval.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o useIntInterval.cmo useIntInterval.ml
ocamlfind ocamlc -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax -p
ackage sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -pac
kage core intInterval.cmo useIntInterval.cmo -o useIntInterval.byte
carbon:v1$
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intInterval.ml > intInterval.ml.depends
```

```
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intInterval.cmo intInterval.ml
ocamlfind ocamlc -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax -p
ackage sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -pac
kage core intInterval.cmo useIntInterval.cmo -o useIntInterval.byte
carbon:v1$ ls
_build/                 intInterval.ml~         useIntInterval.ml
intInterval.ml          useIntInterval.byte@    useIntInterval.ml~
carbon:v1$ ec
carbon:v1$ ls
_build/                 useIntInterval.byte@
intInterval.ml          useIntInterval.ml
carbon:v1$ more _build/intInterval.mli
_build/intInterval.mli: No such file or directory
carbon:v1$ ls _build/
_digests                        ocamlc.where
_log                            useIntInterval.annot
intInterval.annot               useIntInterval.byte*
intInterval.cmi                 useIntInterval.cmi
intInterval.cmo                 useIntInterval.cmo
intInterval.cmt                 useIntInterval.cmt
intInterval.ml                  useIntInterval.ml
intInterval.ml.depends          useIntInterval.ml.depends
carbon:v1$ corebuild intInterval.inferred.mli
ocamlfind ocamlc -i -thread -short-paths -syntax camlp4o -package bin_prot.synta
x -package sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax
-package core intInterval.ml > intInterval.inferred.mli
carbon:v1$ ls
_build/                 useIntInterval.byte@
intInterval.ml          useIntInterval.ml
carbon:v1$ more _build/intInterval.inferred.mli
type intInterval = Interval of int * int | Empty
val is_empty : intInterval -> bool
val contains : intInterval -> int -> bool
val intersect : intInterval -> intInterval -> intInterval
val to_string : intInterval -> string
carbon:v1$ corebuild -clean
carbon:v1$
carbon:v1$
carbon:v1$
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 useIntInterval.ml > useIntInterval.ml.depends
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intInterval.mli > intInterval.mli.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intInterval.cmi intInterval.mli
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
```

```
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o useIntInterval.cmo useIntInterval.ml
+ ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bi
n-annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package s
explib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package cor
e -o useIntInterval.cmo useIntInterval.ml
File "useIntInterval.ml", line 3, characters 21-29:
Error: Unbound constructor IntInterval.Interval
Command exited with code 2.
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 useIntInterval.ml > useIntInterval.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o useIntInterval.cmo useIntInterval.ml
+ ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bi
n-annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package s
explib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package cor
e -o useIntInterval.cmo useIntInterval.ml
File "useIntInterval.ml", line 3, characters 9-27:
Error: Unbound value IntInterval.create
Command exited with code 2.
carbon:v1$ corebuild -clean
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 useIntInterval.ml > useIntInterval.ml.depends
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intInterval.mli > intInterval.mli.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intInterval.cmi intInterval.mli
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o useIntInterval.cmo useIntInterval.ml
+ ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bi
n-annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package s
explib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package cor
e -o useIntInterval.cmo useIntInterval.ml
File "useIntInterval.ml", line 3, characters 9-27:
Error: Unbound value IntInterval.create
Command exited with code 2.
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intInterval.mli > intInterval.mli.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intInterval.cmi intInterval.mli
```

```
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core 
-o useIntInterval.cmo useIntInterval.ml
+ ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bi
n-annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package s
explib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package cor
e -o useIntInterval.cmo useIntInterval.ml
File "useIntInterval.ml", line 3, characters 29-33:
Error: This expression has type 'a * 'b
       but an expression was expected of type int
Command exited with code 2.
carbon:v1$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 useIntInterval.ml > useIntInterval.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core 
-o useIntInterval.cmo useIntInterval.ml
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intInterval.ml > intInterval.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core 
-o intInterval.cmo intInterval.ml
ocamlfind ocamlc -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax -p
ackage sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -pac
kage core intInterval.cmo useIntInterval.cmo -o useIntInterval.byte
carbon:v1$ ./useIntInterval.byte
An interval: ( 3, 4 )
Another interval: ( 3, 7 )
Their intresection: ( 3, 4 )
carbon:v1$ utop
─────────┬──────────────────────────────────────────────────────────┬─────────
         │ Welcome to utop version 1.14 (using OCaml version 4.01.0)! │
         └──────────────────────────────────────────────────────────┘

Findlib has been successfully loaded. Additional directives:
  #require "package";;      to load a package
  #list;;                   to list the available packages
  #camlp4o;;                to load camlp4 (standard syntax)
  #camlp4r;;                to load camlp4 (revised syntax)
  #predicates "p,q,...";;   to set these predicates
  Topfind.reset();;         to force that packages will be reloaded
  #thread;;                 to enable threads


Type #utop_help for help about using utop.

─( 18:00:00 )─< command 0 >──────────────────────────────────────{ counter: 0 }─
utop # #mod_use "intInterval.ml" ;;
module IntInterval :
  sig
    type intInterval = Interval of int * int | Empty
    type t = intInterval
    val create : int -> int -> t
    val is_empty : t -> bool
    val contains : t -> int -> bool
    val intersect : t -> t -> t
    val to_string : t -> string
  end
─( 12:04:22 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "useIntInterval.ml";;
val i1 : IntInterval.t = IntInterval.Interval (3, 4)
val i2 : IntInterval.t = IntInterval.Interval (3, 7)
An interval: ( 3, 4 )
Another interval: ( 3, 7 )
Their intresection: ( 3, 4 )
─( 12:04:33 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # let i = IntInterval.Interval (3,4) ;;
val i : IntInterval.t = IntInterval.Interval (3, 4)
─( 12:04:37 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # #quit
;;
carbon:v1$
```