```
Last login: Mon Jan 30 13:18:31 on ttys024
carbon:SamplePrograms$ cd Sec_01_1\:25pm/
carbon:Sec_01_1:25pm$ 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 # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
─( 13:31:39 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # find_all_by check "dog" m ;;
- : (string * int) list = [("dog", 1); ("dog", 3)]
─( 13:31:48 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # lookup_all "dog" m ;;
- : int list = [1; 3]
─( 13:31:57 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
─( 13:32:30 )─< command 4 >──────────────────────────────────────{ counter: 0 }─
utop # snds (find_all_by check "dog" m) ;;
- : int list = [1; 3]
─( 13:35:16 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_by' : 'a -> ('a * 'b) list -> 'b list = <fun>
─( 13:35:32 )─< command 6 >──────────────────────────────────────{ counter: 0 }─
utop # lookup_by' "dog" m ;;
- : int list = [1; 3]
─( 13:36:59 )─< command 7 >──────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_all' : 'a -> ('a * 'b) list -> 'b list = <fun>
val is_elem_by : ('a -> 'b -> bool) -> 'b -> 'a list -> bool = <fun>
─( 13:37:21 )─< command 8 >──────────────────────────────────────{ counter: 0 }─
utop # is_elem_by streq "dog" ["Hello"; "dog"; "cat"];;
- : bool = true
─( 13:38:52 )─< command 9 >──────────────────────────────────────{ counter: 0 }─
utop # Char.code ;;
- : char -> int = <fun>
─( 13:39:17 )─< command 10 >─────────────────────────────────────{ counter: 0 }─
utop # Char.code 'a' ;;
- : int = 97
─( 13:41:14 )─< command 11 >─────────────────────────────────────{ counter: 0 }─
utop # Char.code 'A' ;;
- : int = 65
─( 13:41:21 )─< command 12 >─────────────────────────────────────{ counter: 0 }─
utop # is_elem_by (fun c i -> i = Char.code c) 67 ['e'; 't'; 'C'; 'f'] ;;
- : bool = true
─( 13:41:24 )─< command 13 >─────────────────────────────────────{ counter: 0 }─
utop # find_all_by (fun c i -> i = Char.code c) 67 ['e'; 't'; 'C'; 'f'] ;;
- : char list = ['C']
─( 13:43:02 )─< command 14 >─────────────────────────────────────{ counter: 0 }─
utop # is_elem_by (=) 5 [1;2;3;4;5;6;7;3;4;5] ;;
- : bool = true
─( 13:43:17 )─< command 15 >─────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_all' : 'a -> ('a * 'b) list -> 'b list = <fun>
val is_elem_by : ('a -> 'b -> bool) -> 'b -> 'a list -> bool = <fun>
val find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
─( 13:46:20 )─< command 16 >─────────────────────────────────────{ counter: 0 }─
utop # find_all_with (fun x -> x = 5) [1;2;3;4;5;4;5;6;7] ;;
- : int list = [5; 5]
─( 13:48:21 )─< command 17 >─────────────────────────────────────{ counter: 0 }─
utop # find_all_with ( (=) 5) [1;2;3;4;5;4;5;6;7] ;;
- : int list = [5; 5]
─( 13:49:05 )─< command 18 >─────────────────────────────────────{ counter: 0 }─
utop # find_all_with (fun x -> x > 5) [1;2;3;4;5;4;5;6;7] ;;
- : int list = [6; 7]
─( 13:49:42 )─< command 19 >─────────────────────────────────────{ counter: 0 }─
```

```
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_all' : 'a -> ('a * 'b) list -> 'b list = <fun>
val is_elem_by : ('a -> 'b -> bool) -> 'b -> 'a list -> bool = <fun>
val find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'a -> 'b list -> 'b list = <fun>
val find_all_by'' : ('a -> 'b -> bool) -> 'a -> 'b list -> 'b list = <fun>
─( 13:50:23 )─< command 20 >────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_all' : 'a -> ('a * 'b) list -> 'b list = <fun>
val is_elem_by : ('a -> 'b -> bool) -> 'b -> 'a list -> bool = <fun>
val find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val find_all_by'' : ('a -> 'b -> bool) -> 'a -> 'b list -> 'b list = <fun>
─( 13:55:27 )─< command 21 >────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_all' : 'a -> ('a * 'b) list -> 'b list = <fun>
val is_elem_by : ('a -> 'b -> bool) -> 'b -> 'a list -> bool = <fun>
val find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val find_all_by'' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
─( 13:59:28 )─< command 22 >────────────────────────────────────{ counter: 0 }─
utop # find_all_by'' (=) 6 [1;2;3;5;6;7;3;6;8];;
- : int list = [6; 6]
─( 14:01:00 )─< command 23 >────────────────────────────────────{ counter: 0 }─
utop # find_all_by'' (=) 6;;
- : int list -> int list = <fun>
─( 14:01:17 )─< command 24 >────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
```

```
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_all' : 'a -> ('a * 'b) list -> 'b list = <fun>
val is_elem_by : ('a -> 'b -> bool) -> 'b -> 'a list -> bool = <fun>
val find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val find_all_by'' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val find_all_with' : ('a -> bool) -> 'b -> 'a list -> 'a list = <fun>
─( 14:01:25 )─< command 25 >─────────────────────────────────────{ counter: 0 }─
utop # #use "find_and_lookup.ml";;
val m : (string * int) list =
  [("dog", 1); ("chicken", 2); ("dog", 3); ("cat", 5)]
val lookup_all : 'a -> ('a * 'b) list -> 'b list = <fun>
val streq : 'a -> 'a -> bool = <fun>
val check : 'a * 'b -> 'a -> bool = <fun>
val find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val snds : ('a * 'b) list -> 'b list = <fun>
val lookup_all' : 'a -> ('a * 'b) list -> 'b list = <fun>
val is_elem_by : ('a -> 'b -> bool) -> 'b -> 'a list -> bool = <fun>
val find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val find_all_by'' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val find_all_with' : ('a -> bool) -> 'b -> 'a list -> 'a list = <fun>
val find_all : 'a -> 'a list -> 'a list = <fun>
val drop_while : 'a list -> ('a -> bool) -> 'a list = <fun>
─( 14:04:49 )─< command 26 >─────────────────────────────────────{ counter: 0 }─
utop # drop_whiledd [4;4;4;5;6;7]
 ((=) 4)  ;;
Error: Unbound value drop_whiledd
Did you mean drop_while?
─( 14:14:04 )─< command 27 >─────────────────────────────────────{ counter: 0 }─
utop # drop_while [4;4;4;5;6;7]
 ((=) 4)  ;;
- : int list = [5; 6; 7]
─( 14:14:41 )─< command 28 >─────────────────────────────────────{ counter: 0 }─
utop #
┌───┬────────────┬─────┬───────────┬──────────────┬───────┬────────┬──────┬────┐
│Arg│Arith_status│Array│ArrayLabels│Assert_failure│Big_int│Bigarray│Buffer│Call│
└───┴────────────┴─────┴───────────┴──────────────┴───────┴────────┴──────┴────┘
```