```
Last login: Mon Jan 30 15:44:06 on ttys028
carbon:~$ c2
carbon:17_Fall_2041$ cd carbon-repos/public-class-repo/SamplePrograms/Sec_10_3\:
35pm/
carbon:Sec_10_3:35pm$ 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_all_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 find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
─( 15:44:39 )─< command 1 >─────────────────────────────────{ counter: 0 }─
utop # #use "find_all_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 find_all_by : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
─( 15:44:45 )─< command 2 >─────────────────────────────────{ counter: 0 }─
utop # find_all_by streq "dog" m ;;
Error: This expression has type (string * int) list
       but an expression was expected of type string list
       Type string * int is not compatible with type string
─( 15:45:37 )─< command 3 >─────────────────────────────────{ counter: 0 }─
utop # fst ;;
- : 'a * 'b -> 'a = <fun>
─( 15:46:01 )─< command 4 >─────────────────────────────────{ counter: 0 }─
utop # #use "find_all_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>
─( 15:48:28 )─< command 5 >─────────────────────────────────{ counter: 0 }─
utop # find_all_by check "dog" m ;;
- : (string * int) list = [("dog", 1); ("dog", 3)]
─( 15:49:24 )─< command 6 >─────────────────────────────────{ counter: 0 }─
utop # lookup_all "dog" m ;;
- : int list = [1; 3]
─( 15:49:34 )─< command 7 >─────────────────────────────────{ counter: 0 }─
utop # snds (find_all_by check "dog" m) ;;
Error: Unbound value snds
Did you mean snd?
─( 15:49:55 )─< command 8 >─────────────────────────────────{ counter: 0 }─
utop # #use "find_all_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>
```
-(15:52:29)-< command 9 >─────────────────────────────────{ counter: 0 }-
utop # snd ;;
- : 'a * 'b -> 'b = <fun>
-(15:54:23)-< command 10 >────────────────────────────────{ counter: 0 }-
utop # snd (1,"4") ;;
- : string = "4"
-(15:54:26)-< command 11 >────────────────────────────────{ counter: 0 }-
utop # snds [(1,"H"); (2,"W")] ;;
- : string list = ["H"; "W"]
-(15:54:49)-< command 12 >────────────────────────────────{ counter: 0 }-
utop # snds (find_all_by check "dog" m) ;;
- : int list = [1; 3]
-(15:55:09)-< command 13 >────────────────────────────────{ counter: 0 }-
utop # (find_all_by check "dog" m) ;;
- : (string * int) list = [("dog", 1); ("dog", 3)]
-(15:55:18)-< command 14 >────────────────────────────────{ counter: 0 }-
utop # find_all_by (fun x y -> x > y) 5 [1;34;56;2;3;5];;
- : int list = [34; 56]
-(15:55:28)-< command 15 >────────────────────────────────{ counter: 0 }-
utop # Char.code ;;
- : char -> int = <fun>
-(15:59:26)-< command 16 >────────────────────────────────{ counter: 0 }-
utop # Char.code 'a' ;;
- : int = 97
-(16:01:04)-< command 17 >────────────────────────────────{ counter: 0 }-
utop # find_all_by (fun c i -> Char.code c > i) 101 ['a'; 'W'; 'r'; '3'] ;;
- : char list = ['r']
-(16:01:09)-< command 18 >────────────────────────────────{ counter: 0 }-
utop # Char.code 'W' ;;
- : int = 87
-(16:02:26)-< command 19 >────────────────────────────────{ counter: 0 }-
utop # find_all_by (=) 5 [1;34;56;2;3;5];;
- : int list = [5]
-(16:02:36)-< command 20 >────────────────────────────────{ counter: 0 }-
utop # #use "find_all_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 find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'a -> 'b list -> 'b list = <fun>
```
-( 16:05:28 )-< command 21 >────────────────────────────────{ counter: 0 }-
utop # #use "find_all_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 find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
```
─( 16:13:26 )─< command 22 >──────────────────────────────────{ counter: 0 }─
utop # #use "find_all_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 find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val flip : ('a -> 'b -> 'c) -> 'b -> 'a -> 'c = <fun>
```
─( 16:15:06 )─< command 23 >──────────────────────────────────{ counter: 0 }─
utop # (>) ;;
- : 'a -> 'a -> bool = <fun>
─( 16:20:21 )─< command 24 >──────────────────────────────────{ counter: 0 }─
utop # (>) 4 3 ;;
- : bool = true
─( 16:21:12 )─< command 25 >──────────────────────────────────{ counter: 0 }─
utop # (>) 4 31 ;;
- : bool = false
─( 16:21:25 )─< command 26 >──────────────────────────────────{ counter: 0 }─
utop # (flip (>)) 4 31 ;;
- : bool = true
─( 16:21:30 )─< command 27 >──────────────────────────────────{ counter: 0 }─
utop # flip ;;
- : ('a -> 'b -> 'c) -> 'b -> 'a -> 'c = <fun>
─( 16:21:41 )─< command 28 >──────────────────────────────────{ counter: 0 }─
utop # #use "find_all_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 find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val flip : ('a -> 'b -> 'c) -> 'b -> 'a -> 'c = <fun>
val compose : ('a -> 'b) -> ('c -> 'a) -> 'c -> 'b = <fun>
File "find_all_lookup.ml", line 48, characters 29-38:
Error: This expression has type int -> int
       but an expression was expected of type int -> char
       Type int is not compatible with type char
```
─( 16:21:58 )─< command 29 >──────────────────────────────────{ counter: 0 }─

```
utop # #use "find_all_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 find_all_with : ('a -> bool) -> 'a list -> 'a list = <fun>
val find_all_by' : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list = <fun>
val flip : ('a -> 'b -> 'c) -> 'b -> 'a -> 'c = <fun>
val compose : ('a -> 'b) -> ('c -> 'a) -> 'c -> 'b = <fun>
-( 16:27:44 )-< command 30 >──────────────────────────────────{ counter: 0 }-
utop # compose Char.code (__(+)_1__) ;;
Error: This expression has type int -> int
       but an expression was expected of type int -> char
       Type int is not compatible with type char
-( 16:27:57 )-< command 31 >──────────────────────────────────{ counter: 0 }-
utop # compose ((+) 1) Char.code ;;
- : char -> int = <fun>
-( 16:28:10 )-< command 32 >──────────────────────────────────{ counter: 0 }-
utop # (compose ((+) 1) Char.code)   'g'  ;;
- : int = 104
-( 16:28:24 )-< command 33 >──────────────────────────────────{ counter: 0 }-
utop # Char.code 'g' ;;
- : int = 103
-( 16:28:35 )-< command 34 >──────────────────────────────────{ counter: 0 }-
utop #
```

Arg	Arith_status	Array	ArrayLabels	Assert_failure	Big_int	Bigarray	Buffer	Call