```
Last login: Wed Apr  4 12:29:16 on ttys008
carbon:$ utop
┌──────────────────────────────────────────────────────────────┐
│ Welcome to utop version 2.0.2 (using OCaml version 4.06.0)! │
└──────────────────────────────────────────────────────────────┘

Type #utop_help for help about using utop.

─( 13:36:46 )─< command 0 >──────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
val subsetsum_exn_on_found : int list -> int list option =
  <fun>
exception KeepLooking
val subsetsum_exn_not_found : int list -> int list option =
  <fun>
val process_solution_exn :
  ('a list -> string) -> 'a list -> 'a list = <fun>
val subsetsum_exn : int list -> int list option = <fun>
val subsetsum_exn_continutation :
  int list -> (int list -> int list) -> int list option = <fun>
val subsetsum_exn_v1 : 'a -> 'b option = <fun>
val subsetsum_exn_first : 'a -> 'b option = <fun>
val subsetsum_exn_print_all : 'a -> 'b option = <fun>
val results : '_weak1 list ref = {contents = []}
val subsetsum_exn_save_all : 'a -> 'b option = <fun>
─( 13:36:46 )─< command 1 >──────────────────────{ counter: 0 }─
utop # subsetsum_exn s ;;
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...
- : int list option = Some [3; -2; 5; -6]
─( 13:36:52 )─< command 2 >──────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
```

```
exception FoundSubSet of int list
val subsetsum_exn_on_found : int list -> int list option =
  <fun>
exception KeepLooking
val subsetsum_exn_not_found : int list -> int list option =
  <fun>
val process_solution_exn :
  ('a list -> string) -> 'a list -> 'a list = <fun>
val subsetsum_exn : int list -> int list option = <fun>
val subsetsum_exn_continutation :
  int list -> (int list -> int list) -> int list option = <fun>
File "search_exceptions.ml", line 170, characters 31-79:
Error: This expression has type int list -> int list
       but an expression was expected of type int list
─( 13:37:15 )─< command 3 >──────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
val subsetsum_exn_on_found : int list -> int list option =
  <fun>
exception KeepLooking
val subsetsum_exn_not_found : int list -> int list option =
  <fun>
val process_solution_exn :
  ('a list -> string) -> 'a list -> 'a list = <fun>
val subsetsum_exn : int list -> int list option = <fun>
val subsetsum_exn_continutation :
  int list -> (int list -> int list) -> int list option = <fun>
val subsetsum_exn_v1 : int list -> int list option = <fun>
val subsetsum_exn_first : 'a -> 'b option = <fun>
val subsetsum_exn_print_all : 'a -> 'b option = <fun>
val results : '_weak2 list ref = {contents = []}
val subsetsum_exn_save_all : 'a -> 'b option = <fun>
─( 13:43:11 )─< command 4 >──────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
val subsetsum_exn_on_found : int list -> int list option =
  <fun>
exception KeepLooking
```

```
val subsetsum_exn_not_found : int list -> int list option =
  <fun>
val process_solution_exn :
  ('a list -> string) -> 'a list -> 'a list = <fun>
val subsetsum_exn : int list -> int list option = <fun>
val subsetsum_exn_continutation :
  int list -> (int list -> int list) -> int list option = <fun>
val subsetsum_exn_v1 : int list -> int list option = <fun>
val subsetsum_exn_first : 'a -> 'b option = <fun>
val subsetsum_exn_print_all : 'a -> 'b option = <fun>
val results : '_weak3 list ref = {contents = []}
val subsetsum_exn_save_all : 'a -> 'b option = <fun>
─( 13:43:28 )─< command 5 >──────────────────────{ counter: 0 }─
utop # subsetsum_exn_v1 s ;;
Here is a solution: [ 1; 5; -6 ]
Do you like it ?
y
Thanks for playing...
- : int list option = Some [1; 5; -6]
─( 13:43:55 )─< command 6 >──────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
val subsetsum_exn_on_found : int list -> int list option =
  <fun>
exception KeepLooking
val subsetsum_exn_not_found : int list -> int list option =
  <fun>
val process_solution_exn :
  ('a list -> string) -> 'a list -> 'a list = <fun>
val subsetsum_exn : int list -> int list option = <fun>
val subsetsum_exn_continutation :
  int list -> (int list -> int list) -> int list option = <fun>
val subsetsum_exn_v1 : int list -> int list option = <fun>
val subsetsum_exn_first : int list -> int list option = <fun>
File "search_exceptions.ml", line 185, characters 48-52:
Error: Unbound value show
─( 13:44:08 )─< command 7 >──────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
```

```
exception FoundSubSet of int list
val subsetsum_exn_on_found : int list -> int list option =
  <fun>
exception KeepLooking
val subsetsum_exn_not_found : int list -> int list option =
  <fun>
val process_solution_exn :
  ('a list -> string) -> 'a list -> 'a list = <fun>
val subsetsum_exn : int list -> int list option = <fun>
val subsetsum_exn_continutation :
  int list -> (int list -> int list) -> int list option = <fun>
val subsetsum_exn_v1 : int list -> int list option = <fun>
val subsetsum_exn_first : int list -> int list option = <fun>
val subsetsum_exn_print_all : int list -> int list option =
  <fun>
val results : '_weak4 list ref = {contents = []}
val subsetsum_exn_save_all : 'a -> 'b option = <fun>
─( 13:47:48 )─< command 8 >──────────────────────{ counter: 0 }─
utop # subsetsum_exn_print_all s ;;
Here is a solution: [ 1; 5; -6 ]
Here is a solution: [ 3; -2; 5; -6 ]
- : int list option = None
─( 13:48:03 )─< command 9 >──────────────────────{ counter: 0 }─
utop # #use "search_cps.ml";;
val show_list : ('a -> string) -> 'a list -> string = <fun>
val sum : int list -> int = <fun>
val process_solution_cps_v1 :
  ('a -> string) -> 'a -> (unit -> 'b) -> (unit -> 'b) -> 'b =
  <fun>
val try_subset_cps_v1 :
  int list -> int list -> (unit -> 'a) -> (unit -> 'a) -> 'a =
  <fun>
val subsetsum_cps_v1 : int list -> unit = <fun>
val process_solution_cps_v2 :
  ('a -> string) -> 'a -> 'b -> 'c -> unit = <fun>
val try_subset_cps_v2 :
  int list -> int list -> 'a -> (unit -> unit) -> unit = <fun>
val subsetsum_cps_v2 : int list -> unit = <fun>
─( 13:48:12 )─< command 10 >─────────────────────{ counter: 0 }─
utop # subsetsum_cps_v1 s ;;
Oh no ....
- : unit = ()
─( 13:57:49 )─< command 11 >─────────────────────{ counter: 0 }─
utop # s ;;
```

```
- : int list = [1; 3; -2; 5; -6]
─( 13:57:55 )─< command 12 >─────────────────────{ counter: 0 }─
utop # #use "search_cps.ml";;
val show_list : ('a -> string) -> 'a list -> string = <fun>
val sum : int list -> int = <fun>
val process_solution_cps_v1 :
  ('a -> string) -> 'a -> (unit -> 'b) -> (unit -> 'b) -> 'b =
  <fun>
val try_subset_cps_v1 :
  int list -> int list -> (unit -> 'a) -> (unit -> 'a) -> 'a =
  <fun>
val subsetsum_cps_v1 : int list -> unit = <fun>
val process_solution_cps_v2 :
  ('a -> string) -> 'a -> 'b -> 'c -> unit = <fun>
val try_subset_cps_v2 :
  int list -> int list -> 'a -> (unit -> unit) -> unit = <fun>
val subsetsum_cps_v2 : int list -> unit = <fun>
─( 13:58:00 )─< command 13 >─────────────────────{ counter: 0 }─
utop # subsetsum_cps_v1 s ;;
Here is a solution:
[ 1; 5; -6 ]
Do you like it?
y
Yeah, we found a solution.
- : unit = ()
─( 13:59:40 )─< command 14 >─────────────────────{ counter: 0 }─
utop # #use "search_cps.ml";;
val show_list : ('a -> string) -> 'a list -> string = <fun>
val sum : int list -> int = <fun>
val process_solution_cps_v1 :
  ('a -> string) -> 'a -> (unit -> 'b) -> (unit -> 'b) -> 'b =
  <fun>
val try_subset_cps_v1 :
  int list -> int list -> (unit -> 'a) -> (unit -> 'a) -> 'a =
  <fun>
val subsetsum_cps_v1 : int list -> unit = <fun>
val process_solution_cps_v2 :
  ('a -> string) -> 'a -> 'b -> 'c -> unit = <fun>
val try_subset_cps_v2 :
  int list -> int list -> 'a -> (unit -> unit) -> unit = <fun>
val subsetsum_cps_v2 : int list -> unit = <fun>
─( 13:59:47 )─< command 15 >─────────────────────{ counter: 0 }─
utop # fun x -> x ;;
- : 'a -> 'a = <fun>
```