```
Last login: Fri Apr  7 13:14:29 on ttys007
carbon:MyCopiesForSecs$ 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 "search_options.ml";;
val gen_subsets : 'a list -> 'a list list = <fun>                        val
gen_subset' : 'a list -> 'a list list = <fun>                          val s :
int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val subsetsum_v1 : int list -> int list option = <fun>
val subsetsum_option_v2 : int list -> int list = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
val is_elem : 'a -> 'a list -> bool = <fun>
val explode : string -> char list = <fun>
val implode : char list -> string = <fun>
val process_solution_option : ('a -> string) -> 'a -> 'a option = <fun>
val subsetsum_option : int list -> int list option = <fun>
─( 13:36:22 )─< command 1 >──────────────────────────────────{ counter: 0 }─
utop # subsetsum_option s ;;
Here is a solution: [ 1; 5; -6 ]                                        Do y
ou like it ?                                                           y

Thanks for playing...
- : int list option = Some [1; 5; -6]
─( 13:36:27 )─< command 2 >──────────────────────────────────{ counter: 0 }─
utop # subsetsum_option s ;;
Here is a solution: [ 1; 5; -6 ]                                        Do y
ou 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]
─( 13:36:49 )─< command 3 >──────────────────────────────────{ counter: 0 }─
utop # subsetsum_option s ;;
Here is a solution: [ 1; 5; -6 ]                                        Do y
ou like it ?                                                           n

Here is a solution: [ 3; -2; 5; -6 ]
Do you like it ?
n
- : int list option = None
─( 13:37:12 )─< command 4 >──────────────────────────────────{ counter: 0 }─
utop # #quit ;;
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 "search_exceptions.ml";;
exception FoundSubSet of int list                                        val
run : 'a -> unit = <fun>                                               File "se
arch_exceptions.ml", line 41, characters 7-10:
Error: Unbound value sum
─( 13:50:49 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]                                     val
sum : int list -> int = <fun>                                          exceptio
n FoundSubSet of int list
val run : 'a -> unit = <fun>
val subsetsum_exn_on_found : int list -> int list option = <fun>
─( 13:50:56 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_on_found s ;;
- : int list option = Some [1; 5; -6]                                    ─( 1
3:51:17 )─< command 3 >──────────────────────────────────────{ counter: 0 }─utop # s
ubsetsum_exn_on_found s ;;
- : int list option = Some [1; 5; -6]                                    ─( 1
3:51:28 )─< command 4 >──────────────────────────────────────{ counter: 0 }─utop # #
use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]                                     val
sum : int list -> int = <fun>                                          exceptio
n FoundSubSet of int list
val run : 'a -> unit = <fun>
val subsetsum_exn_on_found : int list -> int list option = <fun>
exception KeepLooking
val subsetsum_exn_not_found : int list -> int list option = <fun>
File "search_exceptions.ml", line 101, characters 8-15:
Error: Unbound value is_elem
─( 13:55:30 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]                                     val
sum : int list -> int = <fun>                                          val is_e
lem : 'a -> 'a list -> bool = <fun>
val explode : string -> char list = <fun>
val implode : char list -> string = <fun>
exception FoundSubSet of int list
val run : 'a -> unit = <fun>
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 -> string) -> 'a -> 'a option = <fun>
File "search_exceptions.ml", line 126, characters 31-40:
Error: Unbound value show_list
─( 13:55:32 )─< 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 is_e
lem : 'a -> 'a list -> bool = <fun>
val explode : string -> char list = <fun>
val implode : char list -> string = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
```

```
val run : 'a -> unit = <fun>
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 -> string) -> 'a -> 'a option = <fun>
val subsetsum_exn : int list -> int list option = <fun>
─( 13:55:54 )─< command 7 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn s ;;
Here is a solution:                                                        [ 1;
 5; -6 ]                                                                 Do you l
ike 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]
─( 13:56:34 )─< command 8 >──────────────────────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val is_elem : 'a -> 'a list -> bool = <fun>
val explode : string -> char list = <fun>
val implode : char list -> string = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
val run : 'a -> unit = <fun>
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 -> string) -> 'a -> 'a option = <fun>
val subsetsum_exn : int list -> int list option = <fun>
─( 13:56:34 )─< command 9 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_v1 s;;
Error: Unbound value subsetsum_exn_v1
Did you mean subsetsum_exn?
─( 14:02:43 )─< command 10 >─────────────────────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val is_elem : 'a -> 'a list -> bool = <fun>
val explode : string -> char list = <fun>
val implode : char list -> string = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
val run : 'a -> unit = <fun>
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 -> string) -> 'a -> 'a option = <fun>
File "search_exceptions.ml", line 137, characters 23-33:
Error: This function has type int list -> int list -> int list option
       It is applied to too many arguments; maybe you forgot a `;'.
─( 14:02:51 )─< command 11 >─────────────────────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
```

```
val s : int list = [1; 3; -2; 5; -6]
val sum : int list -> int = <fun>
val is_elem : 'a -> 'a list -> bool = <fun>
val explode : string -> char list = <fun>
val implode : char list -> string = <fun>
val show_list : ('a -> string) -> 'a list -> string = <fun>
exception FoundSubSet of int list
val run : 'a -> unit = <fun>
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 -> string) -> 'a -> 'a option = <fun>
val subsetsum_exn : int list -> int list option = <fun>
val subsetsum_exn_continutation :
  int list -> (int list -> int list option) -> 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 : '_a list ref = {contents = []}
val subsetsum_exn_save_all : int list -> int list option = <fun>
─( 14:03:11 )─< command 12 >──────────────────────────────────────{ 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]
─( 14:04:01 )─< command 13 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_first s ;;
- : int list option = Some [1; 5; -6]
─( 14:04:27 )─< command 14 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_all s ;;
Error: Unbound value subsetsum_exn_all
Did you mean subsetsum_exn_v1?
─( 14:04:40 )─< command 15 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_print_all s ;;
Here you go: [ 1; 5; -6 ]
Here you go: [ 3; -2; 5; -6 ]
- : int list option = None
─( 14:04:46 )─< command 16 >──────────────────────────────────────{ counter: 0 }─
utop # results ;;
- : int list list ref = {contents = []}
─( 14:04:59 )─< command 17 >──────────────────────────────────────{ counter: 0 }─
utop # ! results ;;
- : int list list = []
─( 14:06:23 )─< command 18 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_save_all s ;;
[ 1; 5; -6 ]
[ 3; -2; 5; -6 ]
- : int list option = None
─( 14:06:28 )─< command 19 >──────────────────────────────────────{ counter: 0 }─
utop # ! results ;;
- : int list list = [[3; -2; 5; -6]; [1; 5; -6]]
─( 14:06:36 )─< command 20 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;
```

```
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 "subsetsum_cps.ml";;
val show_list : ('a -> string) -> 'a list -> string = <fun>
val is_elem : 'a -> 'a list -> bool = <fun>
val sum : int list -> int = <fun>
File "subsetsum_cps.ml", line 47, characters 21-28:
Error: Unbound value explode
─( 14:13:34 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop #
```

Arg	Arith_status	Array	ArrayLabels	Assert_failure	Big_int	Bigarray	Buffer	Callback