```
Last login: Fri Apr  7 15:23:42 on ttys015
carbon:MyCopiesForSecs$ cd ../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 "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>
─( 15:46:03 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_option s ;;
Here is a solution: [ 1; 5; -6 ]
Do you like it ?
y
Thanks for playing...
- : int list option = Some [1; 5; -6]
─( 15:46:07 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_option 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]
─( 15:46:20 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_option s ;;
Here is a solution: [ 1; 5; -6 ]
Do you like it ?
y
Thanks for playing...
- : int list option = Some [1; 5; -6]
─( 15:46:33 )─< command 4 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_option 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 ?
n
- : int list option = None
```

```
─( 15:46:47 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;
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 "search_exceptions.ml";;
exception FoundSubSet of int list
val run : 'a -> unit = <fun>
File "search_exceptions.ml", line 40, characters 7-10:
Error: Unbound value sum
─( 16:04:01 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "search_exceptions.ml";;
val sum : int list -> int = <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>
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>
─( 16:04:05 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_on_found
s ;;
Error: Unbound value s
─( 16:04:39 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_on_found
s ;;
Error: Unbound value s
─( 16:04:54 )─< 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>
val is_elem : '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>
val subsetsum_exn : int list -> int list option = <fun>
─( 16:05:15 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_on_found
s ;;
- : int list option = Some [1; 5; -6]
```

```
─( 16:05:17 )─< command 6 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn s ;;
Here is a solution:
[ 1; 5; -6 ]
Do you like it?
y
Thanks for playing...
- : int list option = Some [1; 5; -6]
─( 16:05:19 )─< 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>
val is_elem : '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>
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>
─( 16:05:35 )─< command 8 >──────────────────────────────────────{ 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]
─( 16:11:16 )─< command 9 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_first s ;;
- : int list option = Some [1; 5; -6]
─( 16:11:29 )─< command 10 >─────────────────────────────────────{ 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
─( 16:11:36 )─< command 11 >─────────────────────────────────────{ counter: 0 }─
utop # results ;;
- : int list list ref = {contents = []}
─( 16:11:44 )─< command 12 >─────────────────────────────────────{ counter: 0 }─
utop #  ! results ;;
- : int list list = []
─( 16:12:47 )─< command 13 >─────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_save_all s ;;
[ 1; 5; -6 ]
[ 3; -2; 5; -6 ]
```

```
- : int list option = None
─( 16:12:52 )─< command 14 >──────────────────────────────────────{ counter: 0 }─
utop # ! results ;;
- : int list list = [[3; -2; 5; -6]; [1; 5; -6]]
─( 16:13:02 )─< command 15 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_exn_save_all s ;;
[ 1; 5; -6 ]
[ 3; -2; 5; -6 ]
- : int list option = None
─( 16:13:09 )─< command 16 >──────────────────────────────────────{ counter: 0 }─
utop # ! results ;;
- : int list list = [[3; -2; 5; -6]; [1; 5; -6]; [3; -2; 5; -6]; [1; 5; -6]]
─( 16:13:15 )─< command 17 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;
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 "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
─( 16:19:54 )─< command 1 >──────────────────────────────────────{ 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 explode : string -> char list = <fun>
val implode : char 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>
─( 16:20:04 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # subsetsum_cps_v1 [ 1; 3; -2; 5; -6 ]  ;;
Here is a solution:
[ 1; 5; -6 ]
Do you like it?
y
Yeah, we found one
- : unit = ()
─( 16:21:00 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop #
```

Arg	Arith_status	Array	ArrayLabels	Assert_failure	Big_int	Bigarray	Buffer	Callback